

TEKNILLINEN KORKEAKOULU

Elektroniikan, tietoliikenteen ja automaation tiedekunta

Susanna Pöyhönen

LIKKUVAAN MATERIAALIIN SYNKRONOITUVA LEIKKAUS TAAJUUSMUUTTAJASOVELLUKSENA

Diplomityö, joka on jätetty opinnäytteenä tarkastettavaksi diplomi-
insinöörin tutkintoa varten Espoossa 28.12.2009

Työn valvoja: prof. Jorma Luomi

Työn ohjaaja: DI Petri Maksimainen

Tekijä: Susanna Pöyhönen**Työn nimi:** Liikkuvaan materiaaliin synkronoitava leikkaus taajuusmuuttajasovelluksena**Päivämäärä:** 28.12.2009**Sivumäärä:** 72**Tiedekunta:** Elektroniikan, tietoliikenteen ja automaation tiedekunta**Professuuri:** S-81 Sähkökäytöt**Työn valvoja:** Prof. Jorma Luomi**Työn ohjaaja:** DI Petri Maksimainen / ABB Drives

Tässä työssä tutkitaan taajuusmuuttajan käyttämistä leikkausjärjestelmissä, jossa leikkaustyökalu seuraa leikattavan materiaalin liikettä. Liikkeessä suoritettavan leikkauksen tai lyhemmin lentävän leikkauksen tarkoitus on nostaa linjojen suorituskkyä ja parantaa leikkausjäljen laatua. Leikkaaminen materiaalin ehdoilla mahdollistaa sellaisten materiaalien työstön, joiden laatu on aiemmin kärsinyt linjan pysäyttämisestä. Samaa periaatetta voidaan käyttää myös linjoilla, joissa täytetään pulloja tai suoritetaan leimauksia. Leikkaukseen tarvittava liikkeenohjaus tuotetaan taajuusmuuttajalla lisäämällä toiminnot, jotka laajentavat perusliikkeenohjauksen leikkauksen suorittamiseen. Toteutusvaihtoehtoina ovat nokkapyörätoiminnallisuus tai parametriasetuksista laskettavat paikoitustiedot. Parasta menetelmää etsittäessä selvitettiin markkinoilla olevat laitevalmistajat ja heidän tapansa tuottaa leikkauksen liikkeenohjaus. Mukautusmenetelmäksi valittiin parametrisointi useimpien kilpailijoidenkin tapaan, sillä parametrisoinnilla pystytään joustavammin tuottamaan leikkauksen ohjaus ja leikkaamaan tavoitemittaisia leikkauspaloja. Lisäksi kuljettimen palautus alkuasemaan voidaan optimoida usealla tavalla. Leikkaukseen liittyviä erikoisvaatimuksia selvitettiin haastattelemalla asiantuntijoita ja perehtymällä kilpailijoiden menetelmiin. Selvitystyöstä selvisi muun muassa leikkauksen ohjauksen yhteyteen liitettävät oheistoiminnot, monien käytettävien suureiden yksiköt ja suuruusluokat. Kaiken informaation pohjalta luotiin vielä taajuusmuuttajan muuhun ohjaukseen mukautettava lentävä leikkaus -säädön toteutusehdotus lohkoarkkitehtuurina.

Avainsanat: Taajuusmuuttaja, sähkökäyttö, lentävä leikkuri, kilpailevat laitevalmistajat, sovellusohjelmisto

Author: Susanna Pöyhönen

Name of the work: Flying shear as frequency converter application

Date: 28 December 2009

Number of pages: 72

Faculty: Faculty of Electronics, Communications and Automation

Professorship: S-81 Electric drives

Supervisor: Professor Jorma Luomi

Instructor: M.Sc(Tech.) Petri Maksimainen

This thesis studies the use of frequency converters in shearing and cutting processes, where the shearing tool follows the motion of the material. The purpose of shearing on the fly is to maximize the performance of the main production process without interrupting it, and to improve product quality. When processing materials on the fly, it is possible to process materials that usually cannot be stopped during production. This process does not necessarily have to be a shearing process; it may be a filling or stamping process as well. The motion control needed during processing is produced by adding a functionality library to the frequency converter. This added functionality library is used to perform the motion control shearing, cam functions, or parameterized position calculation. Available flying shear applications were investigated to understand the different ways to implement the flying shear control. Based on this investigation, it was determined that makes it possible to implement more flexible shearing control and to cut materials to the desired length. Additionally, it allows for optimization of the carriage return. Based on interviews with experts, and from competitor implementations, special requirements for shearing were discovered. Investigation led to suggested enhancements in flying shear control, and to the use of specific units of measurement and magnitude. This information was then used to produce a function block architecture for a flying shear implementation.

Keywords: frequency converter, electric drive, flying shear, competitive component manufacturers, solution program

Alkulause

Työ on tehty ABB Oy Drivesin Product AC -tulosityksikön tuotekehitysosastolla Helsingissä.

Kiitän professori Jorma Luomea palautteesta ja lukuisista ohjaavista neuvoista sekä ohjaajaani Petri Maksimaista työympäristön järjestämisestä ja opastuksesta työn aikana. Haluan kiittää Juha Kokkosta, Tero Nousiaista ja Mikko Lammia erittäin mielenkiintoisesta aiheesta, joka räätälöitiin juuri minun osaamiseni ja kiinnostukseni mukaan.

Lopuksi haluan kiittää ystäviäni, veljeäni ja erityisesti rakkaita vanhempiani kannustuksesta lopputyön ja koko opiskelun ajan.

Espoossa joulukuun 28. päivänä 2009

Susanna Pöyhönen

Sisällysluettelo

Alkulause.....	4
Sisällysluettelo.....	5
Symboli- ja lyhenneluettelo.....	6
1 Johdanto.....	8
2 Sovellusalueen esittely	10
2.1 Lentävä leikkaus.....	10
2.2 Leikkausmenetelmät.....	11
2.3 Muut sovelluksen käyttökohteet.....	12
2.4 Järjestelmäkäyttö ohjaustasodiagrammissa.....	14
2.5 Markkinalaajuus	16
2.6 Myyntikanava.....	16
3 Laitteen ACSM1 toiminnallisuuden esittely	18
3.1 Liikkeenohjaussäädön periaate	18
3.2 Taajuusmuuttajakäytön ohjausohjelma	19
3.3 Laitteen mukautus halutulle erikoissovellusalueelle	20
3.3.1 Liikeprofiilin luonti ja hallinnointi nokkapyörätoiminnallisuudella	21
3.3.2 Liikeprofiilin luonti parametreilla.....	23
4 Profiilin luonti ja siihen liittyvät piirteet	24
4.1 Nokkapyöräteknologia vastaan muut sovelluksen toteutusvaihtoehdot.....	24
4.2 Kuljettimen kiihdytys	27
4.3 Leikkaus	29
4.4 Leikkausprofiili	30
4.5 Paluuvaiheen profiilit	37
4.5.1 Energiaoptimoitu paluuvaihe	38
4.5.2 Aikaoptimoitu paluuvaihe.....	45
4.5.3 Paluujakson toteutuksen optimointi.....	46
5 Lentävä leikkaus -sovelluksen toteutus ja erikoisvaatimukset.....	49
5.1 Leikkausparametrit.....	49
5.1.1 Leikkauspituus	49
5.1.2 Leikkausaika	49
5.1.3 Yksiköt ja tarkkuus	50
5.1.4 Materiaalihukka ja sahausrako.....	51
5.2 Muita sovellusteknisiä näkökohtia	52
5.3 Sovelluksen vikatilanteet.....	56
5.4 Työsuojausta koskevia näkökohtia	56
5.5 Lentävä leikkaus -säädön periaate.....	57
6 Pohdinnat ja johtopäätökset.....	59
7 Lähteet	61
Liite A: Kilpailijoilla olevat toiminnallisuudet	64
Liite B: Englanninkieliset käännökset toiminnallisuuksille	67
Liite C: Lentävä leikkaus -säädön toteutusehdotus	68

Symboli- ja lyhenneluettelo

Symbolit

a	kiihtyvyys
J	hitausmomentti
P	teho
s	matka tai pituus
t	aika
$t_{[alaindeksi]}$	jaksoon käytettävä aika
T	vääntömomentti
T_a	kiihdyttävä momentti
T_L	liikettä vastustava kuormitusmomentti
v	nopeus
$v_{ret,0}$	vakionopeus
$v_{ret,1}$	vertonopeus
W	energia

Kreikkalaiset symbolit

ω	kulmanopeus
ξ	vakionopeuden ja vertonopeuden suhde

Alaindeksit

<i>acc</i>	kiihtyvyys
<i>C</i>	kuljetin
<i>cut,start</i>	leikkauksen aloitus
<i>dec</i>	jarrutus
<i>error</i>	paikoituksen korjaus
<i>L</i>	linja
<i>length</i>	leikkauspituus
<i>max</i>	paikoituksen korjaus -jakson huippu
<i>pending</i>	tauko
<i>period</i>	jakso
<i>presync</i>	paikoituksen korjaus
<i>profile</i>	profiili
<i>ret</i>	palautus
<i>ret,0</i>	vakionopeus
<i>ret,1</i>	vertonopeus
<i>sync</i>	leikkaus

Lyhenteet

ABB	Asea Brown Boveri
ACSM1	ABB:n kehittämän taajuusmuuttajan tuotenimi
CPU	Central Processing Unit eli keskusyksikkö
DTC	Direct Torque Control eli suora momenttisäätö
GMC	General Motion Control
HMI	Human Machine Interface eli käyttöliittymä koneen ja ihmisen välillä
OEM	Original Equipment Manufacturer eli tuote- ja järjestelmävalmistaja
PC	Personal Computer eli mikrotietokone

PLC

Programmable Logic Controller eli ohjelmoitava
logiikkaohjain

SI

System Integrator

SPC

Solution Program Composer

1 Johdanto

Taajuusmuuttaja on tehoelektroniikkalaite, jolla voidaan muuttaa sähköverkon vaihtosähkön taajuutta ja suuruutta. Taajuusmuuttajakäytöstä puhutaan, kun taajuusmuuttajan puolijohdekytkimien asentoja muuttamalla säädetään ja ohjataan sähkömoottorin akselin pyörimisnopeutta ja vääntömomenttia. Taajuusmuuttajakäyttöjen on tarkoitus auttaa optimoimaan prosesseja ja parantamaan tuotelaatua.

ABB:n taajuusmuuttajavalikoimaa on täydennetty ACSM1-tuotteella, johon on saatavilla nopeus- ja momenttiohjaus tai paikoittava liikkeenohjaus. Paikoittava liikkeenohjaus on tarkoitettu erityisesti vaativiin sovelluksiin koneenrakennusteollisuudessa, jossa se on osana suurempaa järjestelmää järjestelmäkäyttönä. Järjestelmäkäytön ominaisuuksiin kuuluu laaja muunneltavuus ja paras mahdollinen suorituskyky, jotta käytöt saadaan osaksi isompia järjestelmiä.

Uutena kehityssuuntana viime vuosikymmeninä on ollut taajuusmuuttajien oman älyn kasvu sekä hajautetut ohjausratkaisut. Liikkeenohjauksen siirtyessä ylemmältä ohjauksesta taajuusmuuttajaan etuna on nopeampi ja tarkempi liikkeenhallinnan säätö. Samalla kommunikoinnista aiheutunut viive eliminoiduu, kaapeloinnin tarve pienenee ja komponenttikustannukset alenevat. Paraneva suorituskyky laajentaa peruskäytön soveltuvuutta (Leino 2006).

Taajuusmuuttajia on yleisesti käytössä esimerkiksi liukuhihnajärjestelmissä, joissa linjan nopeutta ohjataan taajuusmuuttajan välityksellä, kun linjan tuotetta käsitellään esimerkiksi leikkaamalla. Tavallisesti linja pysäytetään tuotteen leikkaamisen ajaksi ja sitten kiihdytetään ja pysäytetään seuraavaa leikkausta varten. Jatkuva kiihdyttely kuluttaa energiaa ja järjestelmän mekaanisia osia eikä linjan nopeuden muuttaminen ole aina hyvä ratkaisu. Esimerkiksi kuumasta uunista jatkuvalla syötöllä tulevaa materiaalia ei voi pysäyttää sen leikkaamisen ajaksi.

Leikkausta ei ole hyvä suorittaa siten, että linjalla oleva materiaali liikkuu paikoillaan olevan leikkaustyökalun toimiessa. Seurauksena on materiaalihukkaa, terävaurioita ja huono leikkausjälki. Vaihtoehtona onkin leikata materiaalia liukuhihnämateriaalin ehdoilla, jolloin leikkaustyökalua liikutetaan leikkauksen suorittamiseksi linjan nopeudella.

Liikkuvaan materiaaliin synkronoituvaa leikkausta on lähdetty kehittelemään sovelluksena, joka liitetään erikseen osaksi taajuusmuuttajan liikkeenohjausta. Lähtökohtana on ollut käyttää nokkapyörätoiminnallisuutta leikkauksen toteuttamiseen. Tämän työn tavoitteena on tutkia nokkapyörätoiminnallisuuden sopivuutta leikkauksen toteuttamiseen ja niitä erityisvaatimuksia, joita taajuusmuuttajaan kohdistuu leikkausprosessissa.

Toisena tavoitteena työssä on selvittää muut laitevalmistajat, joilla on liikkuvaan materiaaliin synkronoituva leikkaus -sovellus. Heidän toimintaansa tarkkailemalla ja heitä haastatteleamalla saadaan toteutusreferenssiä sekä käsitys vaatimuksista, jotka sovelluksen on täytettävä. Sen lisäksi, että saadaan selville sovellusasiakkaiden tarpeet, saadaan käsitys siitä, minkälaiset menestymisedellytykset sovelluksella on markkinoilla. Sovelluksen nimi liikkuvaan materiaaliin synkronoituva leikkaus on hieman pitkä, joten

käytetään lyhyiden vuoksi termiä lentävä leikkaus. Se on käännös markkinoilla käytettävästä englanninkielestä ilmauksesta ”flying shear”.

Työn luvussa 2 esitellään lentävä leikkaus -sovellusta. Siinä selitetään tarkemmin, mitä lentävä leikkaus tarkoittaa ja mihin kaikkialle sen käyttöä voi laajentaa. Koska työn tarkoitus on selvittää leikkaukseen liittyviä piirteitä ja suunnitella leikkurille kuljetin, tehdään katsaus muutamiin leikkaustyökaluihin, jotka sopivat lentävään leikkaukseen. Luvussa 2 esitellään myös markkinalaajuus, myyntikanava ja sovelluksen liikkeenohjaustoiminnot ohjaustasodiagrammissa.

Työn sisältö kohdistetaan nimenomaan ACSM1-laitteelle, joten luvussa 3 esitellään laitteen säätö ja ohjausohjelman rakenne. Lisäksi esitellään lentävän leikkauksen mukauttaminen osaksi taajuusmuuttajasovellusta. Vaihtoehtoina on jo osittain valmis nokkapyörätoiminnallisuus tai parametrisointi.

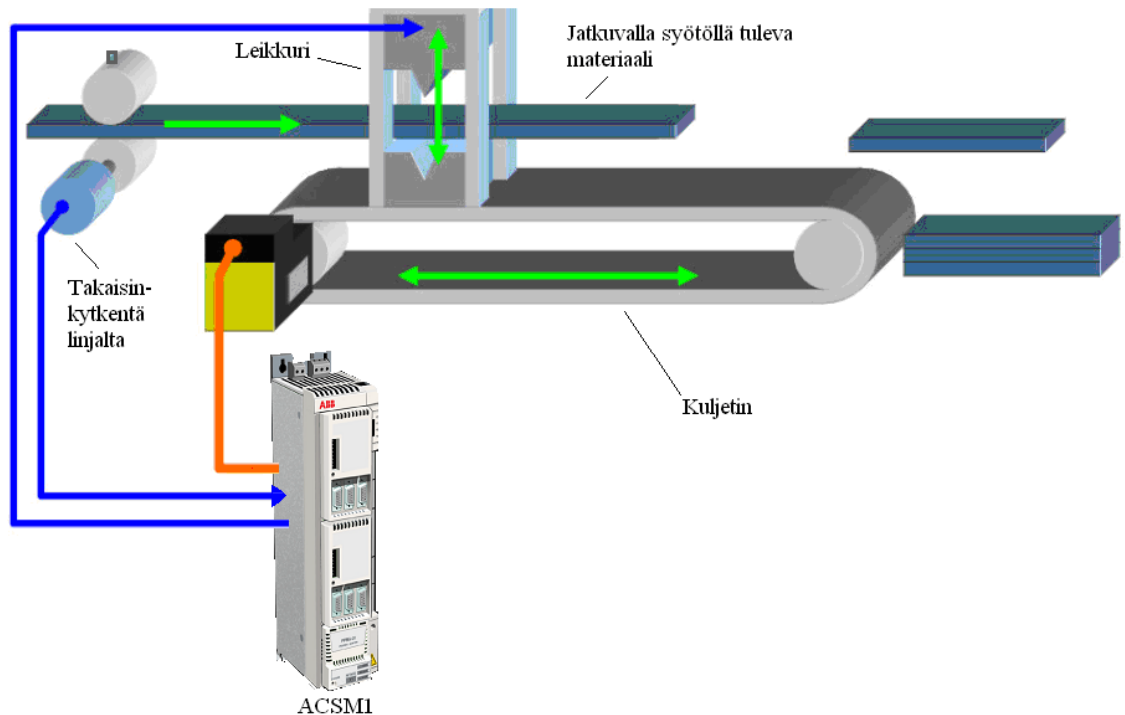
Luvussa 4 esitellään, kuinka lentävä leikkaus -sovellus toteutetaan ja tutkitaan, miten muut laitevalmistajat ovat toteuttaneet sovelluksen. Samassa luvussa käydään tarkemmin läpi leikkausprofiili ja paikoituskaavat. Lisäksi luvussa esitellään kolme paluujakson toteutustapaa.

Kaikki piirteet eivät liikeprofiilien suunnittelussa selviä pelkästään teoreettisella tarkastelulla, joten luvussa 5 on tutkittu ja haastateltu muita laitevalmistajia. Lisäksi mainitaan muutama sana vakavammista vikatilanteista sekä muista sovellusteknisistä ja työsuojelua koskevista näkökohdista. Kun kaikki huomionarvoinen mahdollinen informaatio on saatu kerättyä, esitetään malli säädön toteuttamiseksi osaksi ohjausjärjestelmää.

2 Sovellusalueen esittely

2.1 Lentävä leikkaus

Leikkausprosessissa leikkaustyökalu on asennettu kuljettimeen, joka liikkuu yhdensuuntaisesti leikattavan materiaalin kanssa kuvan 1 tapaan. Leikkurin kuljetin kiihdytetään ja tahdistetaan materiaalin nopeuteen, jonka jälkeen leikkaustyökalu aktivoidaan. Aktivointi saadaan ulkoisen anturin antamasta merkistä tai nopeusmittarin anturin ohi liikkuneen materiaalin pituusinformaation perusteella taajuusmuuttajakäytössä. Leikkauksen jälkeen kuljetin palaa takaisin aloituspisteeseen valmiina seuraavaa leikkausta varten.



Kuva 1. Lentävä leikkaus -sovelluksen luonnos.

Lennossa prosessoinnissa vältetään liukuhihnan pysäyttäminen. Liukuhihna on rakenteeltaan raskaampi kuin kuljetin leikkureineen, joten sen jatkuva pysäyttelyyn ja kiihdyttelyyn tarvitaan suurempi momentti. Toki kuljettimen kiihdyttelyyn ja pysäyttelyyn tarvitaan myös momenttia, mutta liikuteltavan massan määrä on pienempi. Sen seurauksena on järjestelmäkäytön kokonaistehonkulutuksen pieneminen.

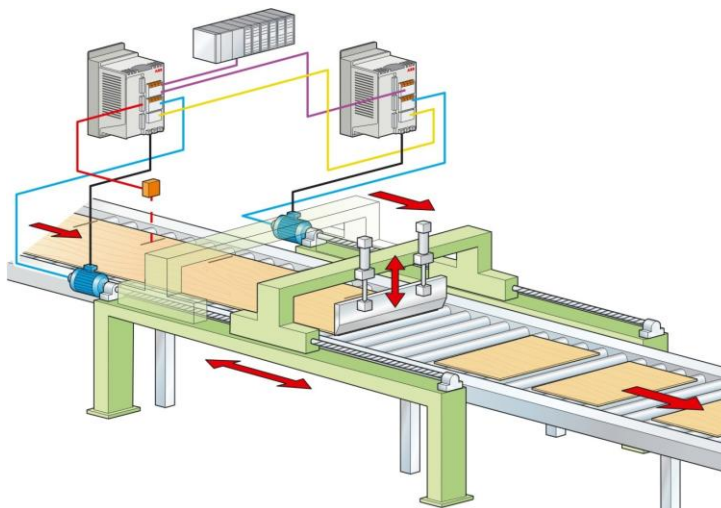
Kuljettimen paikoittamisen tarkoituksena on saada leikkaus suoritettua tarkemmin, jolloin leikkausjäljestä voidaan saada laadukkaampi. Koska linjaa ei missään vaiheessa pysäytetä, leikkausten suorittamiseen saadaan lisää nopeutta. Järjestelmän nopeuden kasvu tarkoittaa samalla myös leikkausjärjestelmän suorituskyvyn kasvua. Lentävässä leikkauksessa linjaa ei pysäytetä, eli voidaan vähentää leikkuujätteen muodostumista. Järjestelmän voidaan siis näin ollen sanoa olevan joustavampi ja tuottoisampi

2.2 Leikkausmenetelmät

Lentävässä leikkauksessa käytettäviä leikkausmenetelmiä voivat olla giljotiinileikkaus ja pyörösahaus. Vannesahaus, laser- ja vesileikkaus voivat olla myös potentiaalisia menetelmiä, mutta niiden soveltuvuus sovellukseen vaatii enemmän selvittelyä. Leikkausmenetelmän valintaan vaikuttaa leikattava materiaali, sen ominaisuudet ja erilaiset leikkaukseen vaikuttavat vaatimukset.

Giljotiinileikkaus

Giljotiinileikkaus tarkoittaa leikkausta, jossa leikkausterä isketään materiaaliin voimalla ja materiaali jakaantuu kahteen osaan. Kuvassa 2 on havainnollistettu leikkausperiaatteen toimintaa. Operointiaika kerrotaan isäntäkäytölle joko kenttäväylän kautta tai asettamalla manuaalisesti isäntäkäyttöön. Tällöin on varmistuttava, että leikkausterä on ylhäällä leikkauksen suorittamisen jälkeen.



Kuva 2. Giljotiinileikkurin toimintaa esittävä kuva (ABB 2008a).

Pyörösahaus

Pyörösahauksessa käytetään pyöreää terää, joka leikkaa pyörimisakseliinsa nähden kohtisuoran railon. Pyörösahan terässä on ulkokehälle sijoitetut hampaat, jotka leikkaavat lastuja terään syötettävästä kappaleesta.

Vannesahaus

Vannesahauksessa teränä käytetään ohutta kehää kiertävää metallivannetta, jonka reuna on hammastettu. Vanteen reunan hampaat leikkaavat kappaletta. Työstettävänä materiaalina voidaan käyttää puuta, metallia tai jotain muuta materiaalia. Vannesahoja voidaan ohjata käyttäen valmiita sahauslaitteen sahausohjelmia, joihin käyttäjä asettaa leikattavan kappaleen muodon, materiaalin ja mittatiedot. Ohjelma optimoi sahauksessa tarvittavat parametrit ja suorittaa sahauksen. Sahaan voidaan asettaa rajat, joita suoruus- ja työstövoimat eivät saa ylittää. Useissa vannesahauslaitteissa on myös teränopeuden säätömahdollisuus taajuusmuuttajasta (Varis ja Tanskanen 2006).

Laserleikkaus

Laserleikkauksessa lasersäde fokusoidaan linssillä materiaalin pintaan 0,1–0,2 mm halkaisijaltaan olevaksi polttopisteeksi, jonka energiatiheys on 1–10 MW/cm². Lämpötila polttopisteessä nousee suureksi ja materiaali sulaa. Muodostunut leikkausjälki on hyvä ja tarkka, ja viimeistelyvaiheen tarve on pieni. Leikkausmateriaalina voidaan käyttää terästä, alumiinia, kuparia, muovia, kumia, puuta ja vaneria (Anon 2009).

Laserleikkausta käytettäessä muodostuu vähemmän pölyä kuin leikkaavissa ja sahaavissa sovelluksissa. Lisäksi huollon tarve on pieni ja muodostunut leikkausjälki on kapea. Toisaalta laserin tehot eivät riitä kovin paksuihin materiaaleihin ja laserin käyttöön liittyy myös riskejä. Leikkauspisteeseen puhalletaan jotain kaasua, esimerkiksi typpeä, joka estää palamisen (Anon 2004). Tämä on tärkeää, sillä pienelle alalle fokusoitu suuri energia aiheuttaa palovaaran.

Vesileikkaus

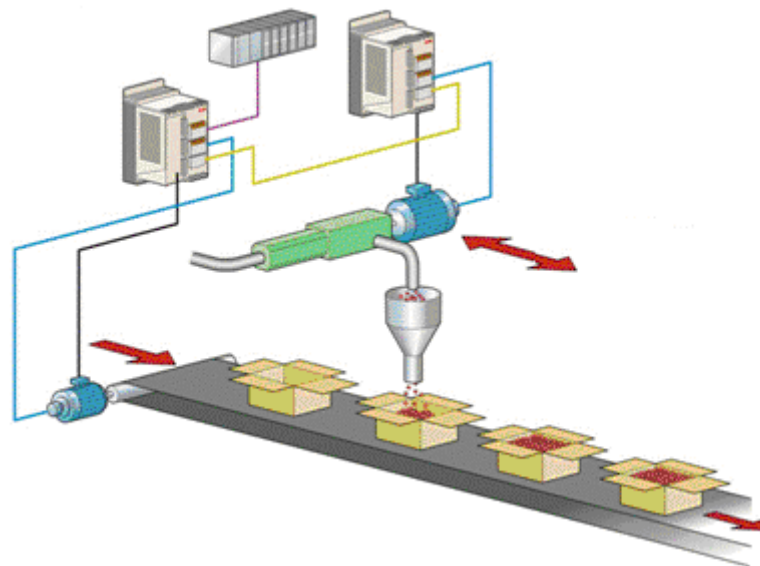
Vesileikkauksessa kohdistetaan paineistettu vesisuihku leikkauskohtaan. Leikkauspaine on 3600 baaria, ja apuaineena käytetään hienoa hiekkaa paksumpia materiaaleja leikattaessa. Vesileikkauksessa leikkauskohtaan ei synny ylimääräistä lämpöä, joten leikkaus on turvallisempi kuin laserleikkaus ja kappaleiden muodonmuutoksilta välttytään. Periaatteessa vesileikkauksella voidaan leikata mitä tahansa vettä kestävästä materiaalista, mutta erityisen hyvin se syö kumia ja vaneria, joiden leikkaamisessa yleensä turvaudutaan sahoihin (Lehtinen 2008).

2.3 Muut sovelluksen käyttökohteet

Pääasiallinen lentävän leikkauksen sovellusalue on jatkuvalla syötöllä tulevan materiaalin leikkaus pienempiin paloihin, kun materiaalin syöttöä ei voida pysäyttää leikkauksen ajaksi. On olemassa myös muita sovelluksia, joissa laite on tahdistettava materiaalin nopeuteen, ja jotka voidaan toteuttaa samalla periaatteella kuin lentävä leikkaus.

Täyttösovellus

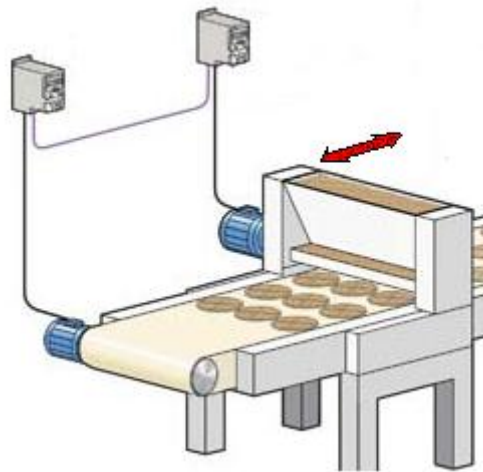
Liukuhihnalla liikkuvia pulloja ja laatikoita voidaan siirtää ja täyttää linjalla pysäyttämättä niitä täytön ajaksi. Täyttäminen tapahtuu siten, että täyttölaite tahdistetaan linjanopeuteen, täyttää pullon tai laatikon, pysäytetään ja palautetaan takaisin kotiasemaan. Kuvassa 3 täytetään laatikoita linjan samalla liikuessa.



Kuva 3. Laatikoiden täyttö (ABB 2008a).

Leimaussovellus

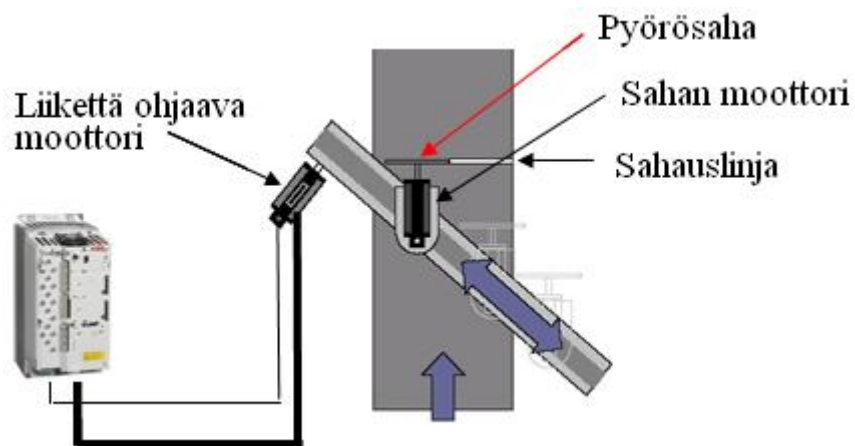
Leimauslaitetta käytetään esimerkiksi, kun halutaan leimata leima laatikon kylkeen. Toinen esimerkki on kuvan 4 tapaan kuvion lyöminen taikinaan, samalla kun linja liikkuu eteenpäin.



Kuva 4. Kuvion leimaaminen taikinaan (ABB 2006).

Leikkaus vinosta kulmasta

Kun materiaalia leikataan vinosta kulmasta, leikkuri liikkuu materiaalivirran yli vinosti kuvan 5 tapaan. Leikkauslinja voi olla $0^\circ \leq \alpha \leq 90^\circ$ kulmassa materiaalivirtaan nähden. Tällöin on huomioitava, että vinottain leikattaessa leikkurin on liikuttava nopeammin kuin yhdensuuntaisesti liikkuva leikkuri. Toiminnallisuus on hyödyllinen sahaussovelluksiin.

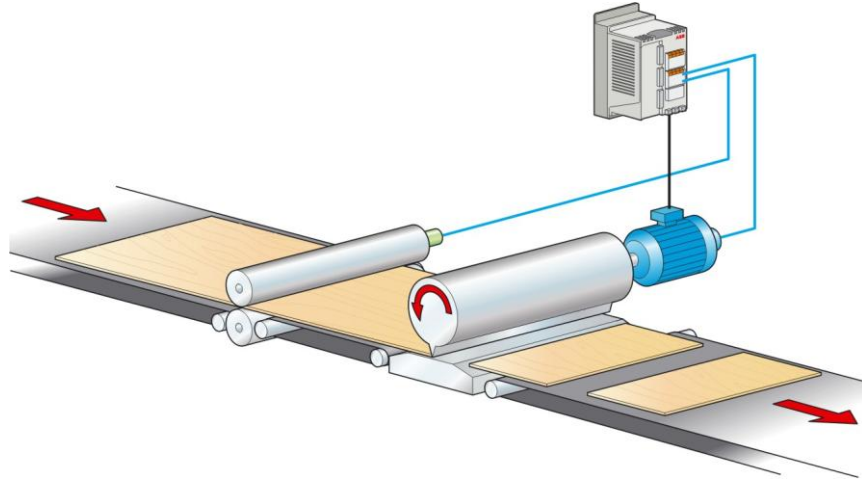


Kuva 5. Leikkauksen suorittaminen vinolla leikkauskulmalla.

Pyörivä leikkuri

Pyörivässä leikkurissa (rotary knife) on pyörivä tela, joissa on yksi tai useampi leikkausterä. Tarkoitus on kohdistaa leikkausterä tai terät oikeaan kohtaan tahdistamalla pyörötelan pyörimisnopeus linjanopeuteen niin, että saadaan halutun pituisia paloja. Pyörivä leikkaus suoritetaan itse asiassa myös liikkeeseen synkronoituvana leikkauksena. Pyörivä leikkuri on kuitenkin rajattu tämän työn ulkopuolelle.

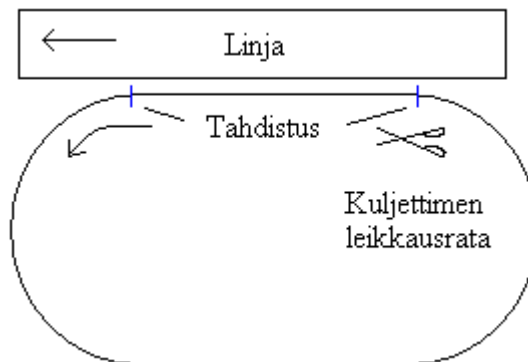
Kuvan 6 tapauksessa linjan yläpuolella on vain yksi pyörötela, mutta pyöröteloja voi olla myös kaksi. Tällöin toinen sijaitsee linjan alapuolella ja toinen linjan yläpuolella. Telojen terät kohdistetaan leikkauksessa samaan aikaan materiaalin läpi.



Kuva 6. Periaatekuva pyörivän leikkurin toiminnasta (ABB 2008a).

Pyörivän ja lentävän leikkauksen hybridi

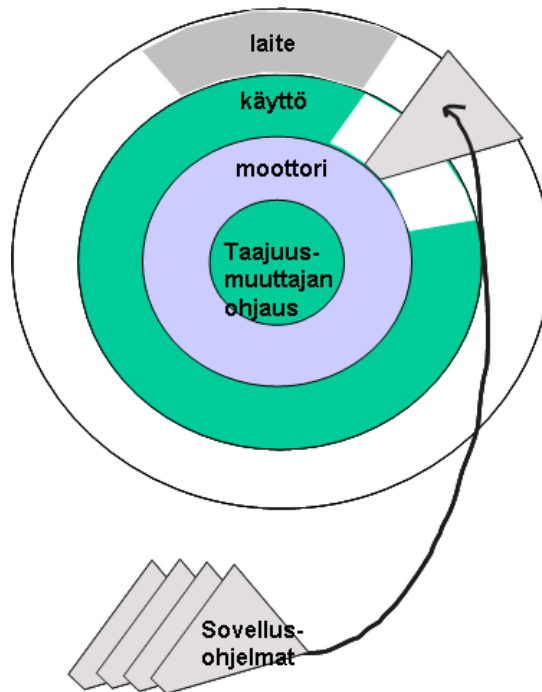
Yhdistämällä lentävän ja pyörivän leikkurin ominaisuuksia saadaan uuden tyyppinen sovellus. Siinä leikkaustyökalun kuljetin liikkuu soikion muotoista rataa pitkin. Leikkuri tahdistuu radan suoralla osuudella linjanopeuteen ja leikkaa, jonka jälkeen kuljetin viipyy radan muissa osuuksissa seuraavaan leikkaukseen asti, kuten havainnollistettu kuvassa 7. Tämän tyyppisessä ratkaisussa pysäytykset ja suunnanvaihdot eliminoituvat, mutta toteuttaminen vaatii tilaa.



Kuva 7. Hybridileikkauksen periaatekuva.

2.4 Järjestelmäkäyttö ohjaustasodiagrammissa

Lentävään leikkaukseen liittyviä komponentteja ja ohjauksen osia voidaan käsitellä järjestelmäkäyttönä. Järjestelmäkäyttö voidaan jakaa neljään kerrokseen, kuten kuvan 8 ohjaustasodiagrammissa on havainnollistettu. Kuvassa sinin kerros kuvaa taajuusmuuttajan ohjausta. Seuraavassa kerroksessa tulevat moottorisäätöön liittyvät taajuusmuuttajan toiminnot. Tämän työn kannalta seuraavat kaksi kerrosta, käyttö ja laite, ovat oleelliset. Kaikkien kerrosten tarkoitus ja sisältö on selitetty lyhyesti seuraavissa kappaleissa.



Kuva 8. Järjestelmäkäytön kuvaus kerroksilla ohjaustasodiagrammissa.

Taajuusmuuttajan ohjaus

Kuvan 8 ohjaustasodiagrammin sisin kerros kuvaa ACSM1-taajuusmuuttajan ohjausta fyysisillä rakenteilla. Laitteen rakenteisiin kuuluvat tasasuuntaaja, välipiiri ja vaihtosuuntaaja. Ohjauksessa syöttävän verkon vaihtojännite tasasuunnataan tasajännitteeksi, joka varastoidaan välipiiriin. ACSM1 on jännitevälipiirillinen taajuusmuuttaja, joten varastointi tapahtuu välipiirin kondensaattoreihin. Välipiirin tasajännite vaihtosuunnataan vaihtojännitteeksi tehopuolijohdekomponenteilla. Taajuusmuuttajan lähdön ohjaaminen tapahtuu kytkemällä tehopuolikomponenteilla sopivasti (Niiranen 1997).

Moottorin säätö – DTC

Taajuusmuuttajalla toteutetaan moottorin tilan ohjaus käyttäen suoraa momenttisäätöä eli DTC:tä (Direct Torque Control), jossa moottoria säädetään suoraan moottorin tilamuuttujilla: vääntömomentilla ja staattorivuolla. Näillä säätösuureilla saatavaa moottorin sähkömagneettista tilaa voidaan käyttää momentin ja nopeuden säätöön (ABB 2008a). Kerros on ensimmäisen kerroksen tapaan ehjä, joka tarkoittaa, että toiminnallisuudet ovat toteutettu vakiona ACSM1-laitteessa ja lisäksi käytettävissä kaikilla moottorityypeillä.

Käyttö

Nopeus- ja momenttisäätötilojen lisäksi moottori voi toimia lukuisissa paikoitustiloissa, joita ovat paikkasäätö, synkronointisäätö ja kotiutus. Nämä lasketaan kuuluvaksi käyttökerrokseen, ja ne on toteutettu järjestelmässä vakiona lähes kokonaan. Joitakin sovelluskohtaisia toimintoja puuttuu, mutta järjestelmän toiminnan kannalta mitään olennaista ei kuitenkaan jää puuttumaan.

Laite – konespesifiset toiminnallisuudet

Ei ole mahdollista toteuttaa yleiskäyttöistä taajuusmuuttajaa, jossa kaikki sovellukset toteutettaisiin yhdessä paketissa koko markkina-alan kattamiseksi. Parametrien eli asettelumuuttaja-arvojen lukumäärä kasvaisi väistämättä suureksi, jos laitteessa tarjotaan kaikki mahdolliset sovellukset ja niiden ominaisuudet. Käyttäjän olisi hallittava lukuisia monimutkaisia parametrikombinaatioita. Ohjelmasta tulee myös

tällöin helposti vaikeasti toteutettava ja hyvin raskas suoritettavaksi CPU:lle (Central Processing Unit) eli ohjelman suorittavalle keskusyksikölle.

Kuvan 8 ohjaustasodiagrammissa uloin kerros, laitekerros, koostuu useista lohkoista, jotka kuvaavat järjestelmäkäytön eri sovellusalueita. Lohkot ovat suhteessa pieniä verrattuna koko diagrammin uloimman kerroksen pinta-alaan. Lisäämällä uloimpaan kerrokseen vain käyttötarkoitukseen sopivat lohkot pienennetään tarvittavaa prosessorikapasiteettia ja tarvittavan muistin osuutta. Jokaisessa lohkoissa on tällöin myös omat konespesifiset toiminnallisuudet, ja parametrien määrä pysyy kohtuullisena sekä järjestelmä käyttäjäystävällisenä.

Sovellusohjelmat

Jotta taajuusmuuttajajärjestelmä saataisiin sovitettua markkina-alan kuhunkin prosessiin, tarvitaan siis vaihtuvia sovellusohjelmia. Sovellusohjelma sisältää kaikki konespesifiset toiminnallisuudet ja parametrit järjestelmäkäytön kannalta. Yksittäinen sovellus vaatii pienen lohkon kaaviokuvan laite- ja käyttökerroksesta. Kun sovelluksia on useita, saadaan katettua kaaviokuvassa koko markkinaliikeala. ACSM1-laitteeseen on jo toteutettu onnistuneita sovelluksia, kuten esimerkiksi kelain-, hissi- ja nosturisovellus. Lentävästä leikkauksesta tehdään ABB:n uusi sovellus markkinoille.

2.5 Markkinalaajuus

ARC:n General Motion Control 2006 raportin (Himanshu 2006) mukaan leikkaussovelluksien osuus liikkeenohjaussäätöjärjestelmien GMC (General Motion Control) markkinasta vuonna 2005 oli 225 miljoonaa Yhdysvaltain dollaria. Se vastaa 4,8 % koko liikkeenohjaussäätöjärjestelmien markkinasta. Liikkeenohjaussäätöjärjestelmiin katsotaan kuuluvaksi toimituksen sisältämät taajuusmuuttajat, liikkeenohjain, moottori, takaisinkytkennät ja kaapelit. Vertailukohdaksi voidaan ottaa esimerkiksi kelainsovellus. Sen osuus liikkeenohjaussäätöjärjestelmistä vuonna 2005 oli 193 miljoonaa dollaria eli 4,1 % (Himanshu 2006). Lentävä leikkaus -sovelluksella oli siis jo vuonna 2005 suurempi markkinaosuus kuin kelainsovelluksella, joka on jo toteutettu ACSM1-laitteeseen.

Saman raportin mukaan vuonna 2010 leikkaussovelluksia odotetaan toimitettavan markkinoille 282 miljoonan Yhdysvaltain dollarin edestä, joka vastaa 4,5 % liikkeenohjaussäätöjärjestelmien markkinasta (Himanshu 2006). Taajuusmuuttajien osuus toimituksesta on arviolta noin 40 % vuonna 2010 eli taajuusmuuttajien markkinaosuus leikkausjärjestelmissä on 112,8 miljoonaa dollaria. ABB:n tavoitteena on saavuttaa tästä 10 %:n markkinaosuus eli 11,3 miljoonaa dollaria (Mäkeläinen 2009). Toisin sanoen voidaan saada 11,3 miljoonan dollarin verran lisää myyntiä, jos lentävä leikkaus -sovellus toteutetaan. Lisäksi voidaan huomioida, että sovelluksen toteuttaminen kasvattaa peruslaitteen myyntiä, sillä laitevalmistaja ostaa todennäköisesti kaikki tuotantolinjan taajuusmuuttajat samalta laitevalmistajalta.

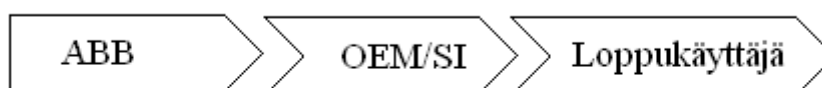
2.6 Myyntikanava

Sovelluksen asiakasryhmä on selvitysten mukaan OEM-toimijat (Original Equipment Manufacturer) ja SI-liiketoimijat (Systems Integrator). Ensimmäinen tarkoittaa kone- ja laitevalmistajia, jotka valmistavat tuotteitaan suurina sarjoina ja myyvät taajuusmuuttajan osana omaa laitettaan (Lönnberg 1995). SI puolestaan tarkoittaa henkilöä tai yritystä, joka on erikoistunut integroituihin järjestelmiin. Näillä ABB:n asiakkailla on tarvittava erikoisosaaminen, mahdollinen kilpailuetu markkinoilla ja välineet toteuttaa sovellus. Siksi on parasta, että nämä toimialat tekevät mahdolliset

muutokset ja lisäykset standardituotteeseen, jotta sovellus saadaan kohdistettua sopivaksi loppukäyttäjän tarpeisiin.

Kuvassa 9 on piirrettyä sovelluksen myyntikanava. Jotta myyntikanava olisi toimiva ja tuottoisa aina ABB:ltä loppukäyttäjälle asti, ABB:n kannattaa ottaa selvää asiakkaiden, kuten myös loppukäyttäjien käyttämistä teknologioista, järjestelmistä, liiketoiminnoista ja tarpeista. Tulevaa toimintaympäristöä tarkkailemalla on mahdollista kartoittaa sovelluksen tarpeita ja ymmärtää loppukäyttäjien näkökulmaa, ja esiin voi nousta myös joitakin ongelmia, jotka voidaan hyvissä ajoin eliminoida (Artto 2008).

OEM-asiakkaiden vaatimukset taajuusmuuttajavalmistajille ovat korkeat. Tuotteet on räätälöitävä täsmälleen heidän koneisiinsa ja laitteisiinsa sopiviksi. Toimiva asiakassuhde heidän kanssaan on ennakkoehto liiketoiminnan aloittamiselle. Tämä takaa sen, että kuvan 9 myyntiketjun viimeisetkin palaset asettuvat arvoketjuun ja näin luontevasti tuottavat koko ketjulle arvoa.



Kuva 9. Mahdollinen myyntikanava.

3 Laitteen ACSM1 toiminnallisuuden esittely

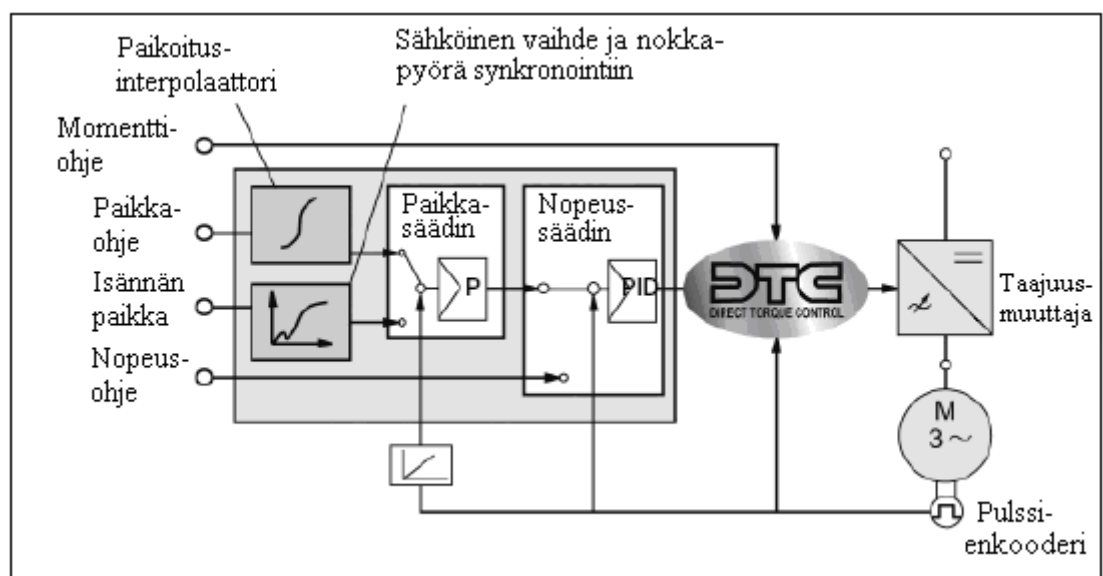
3.1 Liikkeenohjaussäädön periaate

Taajuusmuuttajakäytön tavoitteena on energian siirtäminen säästeliäästi sähköverkosta prosessiin ja luotettavan ohjauksen tuottaminen. Ohjaus perustuu ohjearvoihin ja laitteessa muodostettuun malliin, joiden perusteella muodostetaan tarvittavat laitteen ohjaussignaalit halutun ilmiön saavuttamiseksi. Tavoitteen saavuttamiseen tarvitaan myös säätöä, jolloin laitteen lähtösuureita verrataan ohjearvoihin. Näiden kahden erotuksen eli säätöpoikkeaman perusteella muutetaan ohjaussignaaleja säätöalgoritmin mukaisesti siten, että säätöpoikkeama pienenee.

Ohjauksessa ja säädössä käytetään skalaarisäätöä, skalaariohjausta, vektorisäätöä ja DTC:tä, joista viimeisintä käytetään yleisesti ACSM1-taajuusmuuttajassa. DTC:n etuna ovat nopea momenttivaste ja nopeustarkkuus, jotka takaavat hyvät dynaamiset suoritusarvot. Nämä parantavat prosessien hallittavuutta ja tuotteiden laadun yhdenmukaisuutta.

DTC muodostuu kahdesta pääosasta: momentinsäätöpiiristä ja nopeudensäätöpiiristä. Käytännössä säädetään jompaakumpaa suureista, jolloin kyse on momentinsäädöstä tai nopeudensäädöstä. Kun käyttö toimii momentinsäätötilassa, nopeus säätöy kuorman mukaan. Vastaavasti taas nopeudensäätötilassa momentti säätöy kuorman mukaan (ABB 2008a).

ACSM1-laitteessa on myös paikoittava liikkeenohjaus, jossa kuorma paikoitetaan yksittäisellä akselilla alkuasennosta määritellyyn kohdepaikkaan paikkaohjeen mukaan. Paikkasäädön ohjeet siirretään sopivassa muodossa eteenpäin nopeussäätimelle kuvan 10 mukaan. Synkronointisäätöä käytetään kahden tai useamman mekaanisen järjestelmän tahdistamiseen. Tämä säätö vastaa paikkasäätöä, mutta synkronointisäädössä paikkaohje otetaan isäntänä toimivasta liikkuvasta kohteesta. Tällöin on aina käytettävä paikan takaisinkytkentää kuorman paikan oloarvon määrittämiseksi.



Kuva 10. Liikkeen säätösilmukka (ABB 2008a).

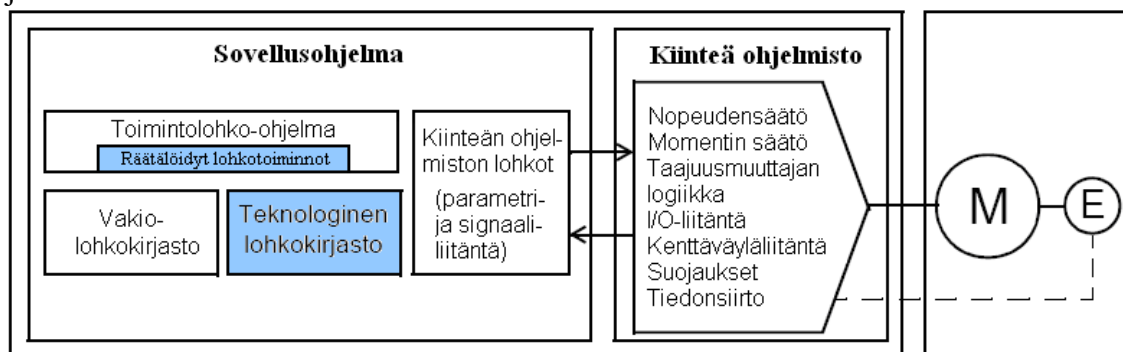
Isäntäkäytöltä saatavaa paikan ohjetta voidaan vielä muuttaa sähköisellä vaihteella ja nokkapyörätoiminnallisuudella (cam-toiminnallisuus). Sähköisellä vaihteella etäisyys voidaan korjata haluttuun mittakaavaan ja nokkapyörätoiminnallisuudella voidaan hallinnoida hyvin epälineaarisia riippuvuuksia ensisijaisen isäntäkäytön ja orjakäytön välillä (ABB 2008b).

3.2 Taajuusmuuttajakäytön ohjausohjelma

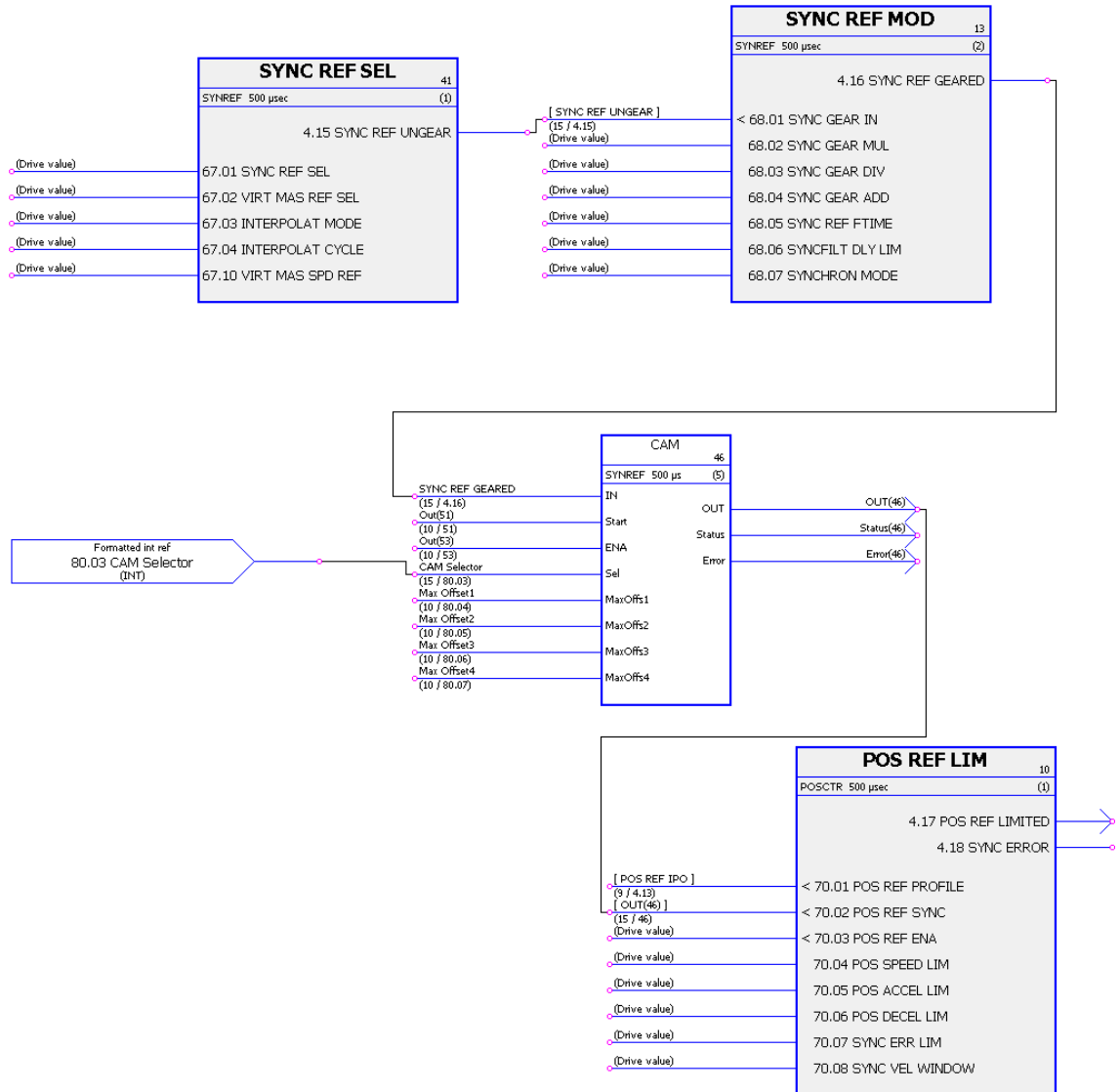
Taajuusmuuttajakäytön ohjausohjelma voidaan jakaa säädön kannalta kahteen osaan. Toinen osa koostuu kiinteästä ohjelmistosta (firmware) ja toinen osa sovellusohjelmasta kuvan 11 tapaan. Kiinteällä ohjelmistolla toteutetaan parametrien hallinnointi ja sovellusohjelmalla lohko-ohjelmointi.

Kiinteän ohjelmiston parametriohjelmoinnilla suoritetaan laitekohtaiset pääohjaustoiminnot, kuten nopeuden ja momentin säätö, taajuusmuuttajan logiikkatoiminnot, tiedonsiirtotoiminnot sekä suojaustoiminnot. Valmiit säätökonfiguraatiot ja asetukset ovat valmiiksi tallennettuna taajuusmuuttajan pysyväismuistiin, ja säätösuureiden tarkkailu on mahdollista käyttöönototyökalujen avulla. Koska parametreja käytetään kiinteän ohjelmiston toimintalohkojen tuloina, parametreja on mahdollista muokata myös sovellusohjelmalla.

Lohko-ohjelmointi eli lohkojen hallinta suoritetaan PC-pohjaisella sovellusohjelmalla DriveSPC (Solution Program Composer). Ohjelmassa olevat lohkot toimivat käyttäjärajapintana, jossa jokaisen toimintalohkon taakse on piilotettu laskenta-algoritmit ja muut tarvittavat informaation käsittelyt eri toiminnoille. Näiden muokkaaminen ja laajennus on mahdollista sovellusohjelmasta käsin. Lohkot ja niihin liittyvät parametrit on sijoitettu DriveSPC:ssä erillisille välilehdille kuvan 12 tapaan, jotta niiden hallinta olisi selkeää.



Kuva 11. Taajuusmuuttajan ohjausohjelman rakenne, jossa siniset osat tarkoittavat käytön laajentamista osaksi sovellusta.



Kuva 12. DriveSPC:n logiikkaohjelmakoodia, jossa harmaat lohkot ovat kiinteän ohjelmiston lohkoja ja lisätyt valkoiset lohkot laajentavat laitteen sovellukselle. Kuvan tapauksessa toteutetaan nokkapyörätoiminnallisuus cam-lohkokolla.

Sovellusohjelmassa on kolmenlaisia toimintalohkoja: kiinteän ohjelmiston toimintalohkoja, vakio-toimintalohkoja ja teknologialohkoja. Kiinteän ohjelmiston toimintalohkot ovat osa taajuusmuuttajan kiinteää ohjelmistoa, ja niitä käytetään liitännänä sovellusohjelman ja kiinteän ohjelmiston välillä. Vakio-toimintalohkoja (esim. ADD, AND) käytetään suoritettavan sovellusohjelman luomiseen. Lohkot perustuvat IEC-61131-3-standardiin, ja lohkojen kirjasto kuuluu vakiona laitteen toimitukseen. Luotaessa eri sovellustyyppettä käytetään teknologisia toimintalohkojen kirjastoja. Teknologia-kirjastot sisältävät kunkin sovelluksen vaatimat toiminnot ja parametrit (ABB 2008b).

3.3 Laitteen mukautus halutulle erikoissovellusalueelle

Kohdassa 2.4 tuotiin esiin, että taajuusmuuttajalaitteessa on vakiona kiinteä ohjelmistoversio, joka on riippumaton sovellusohjelmista. Kun laite halutaan mukauttaa halutulle erikoissovellukselle, sovelluksen vaatimat komponentit ladataan erikseen käyttöön. Käytännössä se tarkoittaa sitä, että aiemman kuvan 11 sovellusohjelmaa laajennetaan lisäämällä laitteeseen sinisellä värillä värjättyt osat: osa toimintalohko-ohjelmaa eli räätälöidyt lohkotoinnot ja teknologia-kirjasto.

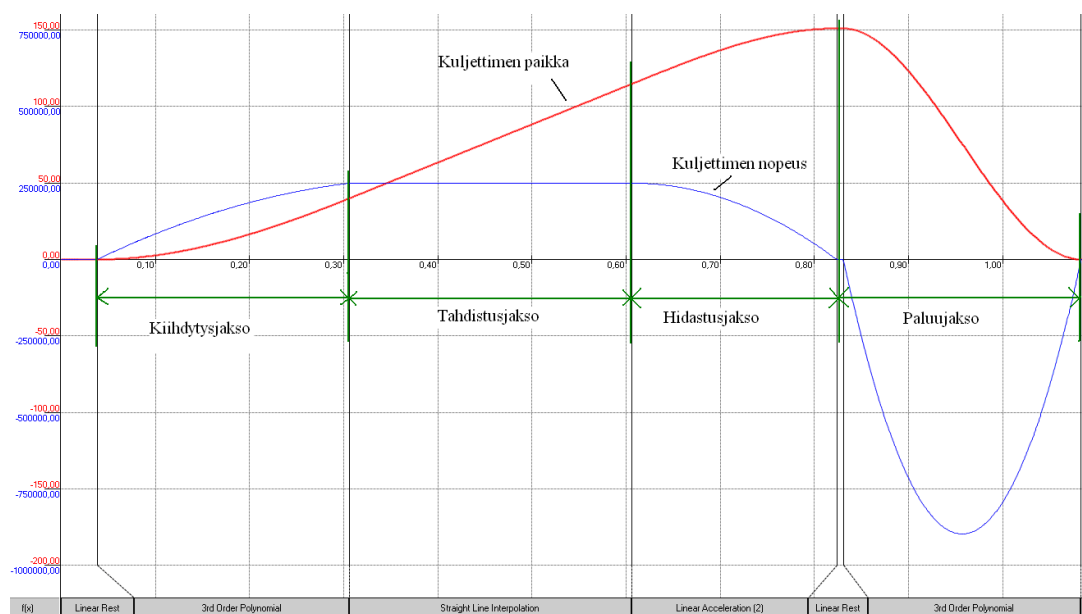
Teknologiakirjaston avulla sovellusohjelmaan voidaan lisätä teknologialohkoja erityisiä käyttösovelluksia varten. Niiden tarkoituksena on lisätä toimitukseen sovellusparametrit jo laitteessa olevien kiinteän ohjelmiston parametrien lisäksi. Räätelöidyt lohko-toiminnot ovat vakiotoimintolohkoista rakennettuja sovellusspesifisiä lohkokokonaisuuksia. Se, minkälaisia lohkoja lentävä leikkaus -sovellukseen kaikkiaan tarvitaan, selviää työn aikana.

3.3.1 Liikeprofiilin luonti ja hallinnointi nokkapyörätoiminnallisuudella

Tällä hetkellä käyttöön ladattava teknologiakirjasto sisältää valmiina cam-lohkon. Epälineaaristen riippuvuuksien luontiin ja suunnitteluun käytetään graafista PC-työkalua DriveCam. Työkalulla luotavilla nokkapyöräprofiileilla on mahdollista tarkkailla visuaalisesti paikkaa, nopeutta, kiihtyvyyttä ja kiihtyvyyden muutosta. Nokkapyöräprofiili voidaan jakaa useisiin jaksoihin, ja kullekin jaksolle on valittavissa 20 erilaista matemaattista laskentatoimintoa, kuten viiva ja korkeamman asteen polynomisia toimintoja. Muutosten tekeminen profiileihin onnistuu käyttöliittymästä suoraan graafisia käyriä muuttamalla.

Yksi nokkapyörätoiminnallisuus voi sisältää kaikkiaan 32 eri liikeprofiilia, jotka jokainen tallennetaan cam-tilukkuun. Profiilin tilukointi tapahtuu jakamalla profiilin x-akseli maksimissaan 4096 tasaväliseen pisteeseen. Jokainen piste käydään läpi ja y-akselin tiedot tallennetaan cam-tilukkuun, joka sitten ladataan orjakäyttöön. Orjakäyttö interpoloi pisteistä profiiliin, vertaa sitä isäntäkäytön nopeus- tai paikkaohjeeseen ja muuttaa liikerataansa epälineaarisesti suhteessa isäntäkäyttöön. Profiileja voidaan luoda myös tilukko-ohjelmilla, kuten Excelillä. Tällöin tilukko-ohjelman perättäiset 4096 alkia ladataan orjakäyttöön ja niistä muodostetaan profiili.

Kuvassa 13 on pelkistetty erään taajuusmuuttajakäyttöön ladattavan lentävää leikkausta esittävän neljän jakson nokkapyöräprofiili, joka vastaa yhtä leikkaus- tai sahaustoimintoa. Profiilissa leikkausterän kuljetin kiihdytetään ensin linjanopeuteen, leikkauksen aikana kuljetimen nopeus on tahdistettu linjanopeuteen, leikkauksen jälkeen kuljetin pysäytetään ja viimeisenä on paluu takaisin lähtöpisteeseen. Erilaisia profiileja, ei pelkästään lentävä leikkaus -profiileja, voidaan luoda rajaton määrä.



Kuva 13. DriveCam-työkalulla tehty visuaalinen lentävän leikkauksen nokkapyöräprofiili, jossa on kuvattu punaisella paikan ja sinisellä nopeuden liikeradat neljässä jaksossa.

Cam-teknologialohkoon kuuluu myös joitakin käyttöliittymästä valittavia cam- taulukoiden käsittelyyn liittyviä toimintoja, kuten suhteellisuus, jatkuvuus ja automaattinen siirtyminen. Toimintojen tarkemmat kuvaukset on taulukoitu taulukkoon 1. Muita asetuksia nokkapyörätoiminnallisuuden käyttöön on paikka-akselin valinta. On määritettävä, onko liike lineaarista vai pyörivää liikettä. Linearisessa liikkeessä on liikuttava paikkainformaation minimi- ja maksimirajojen sisällä. Pyörivässä liikkeessä paikoitus tapahtuu 0 ja 1 kierroksen välillä eli jokaisen kierroksen jälkeen paikan laskenta aloitetaan alusta. Jotta yli kierroksen pituiset leikkauspalat saadaan leikattua, on paikka-akseliksi parempi valita lineaarinen liike.

Orjakäyttö toimii nokkapyörätoiminnallisuudessa synkronointisäädössä, jolloin paikkaohje on aina 0 ja 1 kierroksen sisällä. Käytännössä paikka-akseliksi on kuitenkin siis aina ABB:n laitteessa valittava pyörivä liike kaikkialle taajuusmuuttajakäytön asetuksiin. Tilannetta voidaan korjata käyttämällä nokkapyörätoiminnallisuuden tarjoamaa offset-toimintoa, jolla voidaan venyttää nokkapyöräprofiilin pituutta. Se ei kuitenkaan pysty ylikirjoittamaan synkronointisäädön paikkatietoa. Toinen vaihtoehto on muuttaa linjan paikkalaskenta esimerkiksi viiteen kierrokseen yhden sijasta. Jos linjan paikkalaskennan paikoitusväliä muutetaan, sen täytyy kuitenkin vastata täsmälleen leikkausprofiilin pituutta.

Jotta yli kierroksen pituiset leikkauspituudet saadaan nokkapyörätoiminnallisuuden normaalilla paikkalaskennalla suoritettua, on liikeprofiili jaettava osiin ja kukin osa sijoitettava useaan peräkkäiseen cam- taulukkoon, jotka sitten ajetaan peräkkäin. Liikeprofiilia luodessa eteen voi tulla hahmottamisongelmia. Jos profiilin luonti ei kerralla onnistu, niin profiilin muokkaaminen voi olla hankalaa johtuen epäloogisesta paloittelusta. Tällöin on parempi luoda kokonaan uusi liikeprofiili. Jos leikkauspituus on niin suuri, että se vastaa 32 kierrosta, joudutaan käyttämään kaikki 32 cam- taulukkoa pelkästään yhteen profiiliin. Leikkauspituuden edelleen kasvaessa leikkauksia ei voida suorittaa.

Toisaalta nokkapyörätoiminnallisuudessa koko leikkausprofiili voidaan asettaa yhteen cam- taulukkoon, jos käytetään juuri sopivaa kuorman välityssuhdetta. Välityssuhteen määrittäminen voi olla tarkkuuden takia hankalaa, ja todennäköisesti samaa välityssuhdetta ei voi käyttää seuraavassa suoritettavassa cam- taulukossa tai edes samassa cam- taulukossa, jos linjanopeus muuttuu huomattavasti kesken prosessin. Jos välityssuhde ohjelmoidaan nopeudenmuutosten mukaan vaihtelevaksi, on mahdollista käyttää yhtä leikkausprofiilia leikkauksen suorittamiseen (Siemens 2007).

Taulukko 1. Cam- taulukoiden käsittelyyn liittyviä toimintoja.

Suhteellisuus	Suhteellisessa tilassa orjakäyttö asettaa paikoitustiedon nolla-aseman nolnaan, kun cam-lohkon käyttö aloitetaan. Absoluuttisessa tilassa nolla-asema tiedetään eikä alkuasetuksia palauteta, kun cam-lohkojen käyttö aloitetaan.
Toistettavuus	Määritetään, suoritetaanko tiettyä nokkapyöräprofiilia kerran vai jatkuvasti uudestaan.
Automaattinen siirtyminen	Seuraava nokkapyöräprofiili suoritetaan automaattisesti, kun edellinen nokkapyöräprofiili on suoritettu loppuun. Jos viimeinen nokkapyöräprofiili on määritetty siirrettäväksi automaattisesti, suoritetaan seuraavaksi ensimmäinen nokkapyöräprofiili.

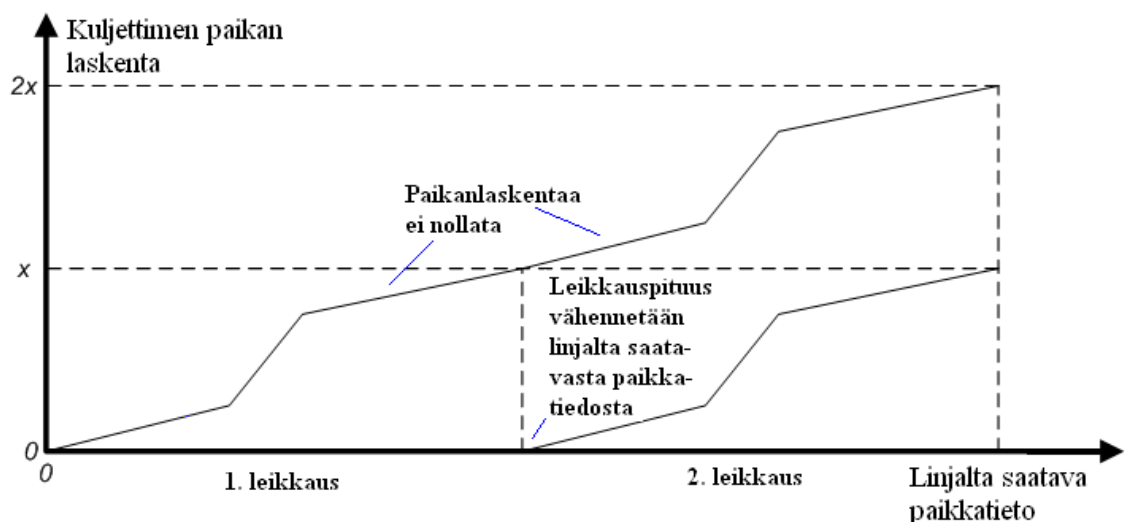
3.3.2 Liikeprofiilin luonti parametreilla

Toinen vaihtoehto epälineaarisen liikeradan muodostamiseen on lisätä toimintalohko-ohjelmaan kokonaan uusi teknologiakirjasto, joka sisältää tarvittavat lohkot ja toimituksen tarvitsemat sovellusparametrit jo laitteessa olevien kiinteän ohjelmiston parametrien lisäksi. Näiden käyttäjän asettamien parametritietojen pohjalta ohjelman säätimet luovat kuvan 13 mallisen kuljettimen leikkausprofiilin. Kuten nokkapyörätoiminnallisuudessaakin, parametrisoitavassa toiminnossa liikeprofiili jaetaan loogisiin jaksoihin. Ohjelmisto ilmoittaa aina muille säätimen osille signaaleilla, mikä on kullakin hetkellä käytössä oleva jakso ja mitkä ovat lasketut kriittiset parametrit.

Jotta välttytään tilanteilta, jossa leikkausta ei kyetä toteuttamaan, on tarkistettava profiilin toteuttamismahdollisuus käytännössä. Samalla tehdään profiilin optimointi, jonka tarkoitus on minimoida energian ja ajan käyttöä. Mikäli varmistustoiminnot onnistuvat, leikkaus voidaan suorittaa. Liikeprofiilin muokkaus ja uudelleenmäärittäminen onnistuvat parametritietoja muuttamalla.

Parametrisoitavassa profiilin luonnissa ongelmaksi muodostuu todellisen paikan seuranta. Leikkausoperaation aikana linjalta tulevaa paikkatietoa eli synkronointireferenssiä ei voi nollata kesken ilman, että paikkatietoa hukkaantuu. Tämä voidaan huomioida lisäämällä leikkauspituus synkronointireferenssiin tai vähentämällä suoritettu leikkauspituus siitä, kuten kuvassa 14 on havainnollistettu. Paikka-arvon laskennassa tulee vain huomioida, ettei synkronointireferenssin arvo kasva 32 768 yksikköön asti. Tämän jälkeen paikka-arvonlaskenta aloitetaan $-32\,768$:sta. Ohjelmaa suoritettaessa ei välttämättä saavuteta maksimaalista referenssiä, mutta varmuudeksi on hyvä tällaisessa tapauksessa kehittää jokin logiikka, sillä leikkaussäätö voi muutoin mennä sekaisin.

Leikkauksen liikeprofiileja voi olla useampi, jos niiden luonti toteutetaan parametrisoimalla. Jokainen erikoisempi profiili on ohjelmoitava erikseen käyttäjärajapintojen alapuolelta tapauskohtaisesti. Ohjelmasta tulee tällöin helposti sotkuinen ja vaikeaselkoinen, sillä jokainen eri liikeprofiili tarvitsee omat parametrinsa. Tarve erityiselle profiilityypille tulee todennäköisesti vastaan, jos leikkaussovellusta käytetään johonkin muuhun käyttötarkoitukseen, kuten pullontäyttöön tai leimaukseen.



Kuva 14. Paikan laskentaa havainnollistava kaavio, jossa linjalta tuleva paikkatieto lisätään sellaisenaan synkronointireferenssiin. Toinen vaihtoehto on nollata kuljettimen paikkatieto linjan paikkatiedosta jokaisen leikkauksen jälkeen.

4 Profiilin luonti ja siihen liittyvät piirteet

Luvussa 3 esiteltiin vaihtoehtoiset tavat toteuttaa sovellus. Tässä luvussa esitellään perustellusti paras menetelmä toteuttaa sovellus. Ohjelman toteutukseen ei vielä puututa, sillä on ensin selvítettävä tarkemmin profiilien käyttäytyminen. Liikeprofiilin luonnissa ja valinnassa on useita huomioon otettavia seikkoja, kuten ohjelman toteutustapa, tarvittavat esiasennukset, leikkauksen käynnistävä valinta ja kuljettimen kiihdytys linjanopeuteen. Lisäksi esitellään kolme kuljettimen palautuksen optimointitapaa.

4.1 Nokkapyöräteknologia vastaan muut sovelluksen toteutusvaihtoehdot

Taulukkoon 2 on koottu laitevalmistajien tavat toteuttaa lentävä leikkaus -sovellus. Nähdään, että vain Yaskawa on toteuttanut sovelluksen nokkapyörätoiminnallisuudella (Yaskawa 2006). Todennäköistä on, että muut ovat käyttäneet nokkapyörätoiminnallisuutta CT:n (CT 2001) tapaan eli parametritietojen pohjalta luodaan liikeprofiilitaulukko, joka muutetaan nokkapyörätoiminnallisuudeksi. Joillakin kilpailijoilla puolestaan näyttää olevan nokkapyörätoiminnallisuus sovellusvalikoimassa, mutta se on selvästi erillinen sovellus, joka ei liity lentävään leikkaukseen. Liitteen A taulukoihin on koottu ominaisuuksia, joita on kilpailijoiden laitteissa. Siitä selviää ne kilpailijat, joilla on nokkapyörätoiminnallisuus toimintotarjonnassaan. Liitteessä B on toiminnallisuudet vielä englanniksi.

CT:n toteutuksessa pistää silmään myös taulukkoalkioiden määrä, 10 kappaletta. Jos CT pärjää kymmenellä taulukkoarvolla, herää kysymys, onko laitteeseen toteutettu nokkapyörätoiminnallisuus maksimaalisella 4096 pisteellä kenties ylimitoitettu? Toisaalta laitteeseen toteutetun nokkapyörätoiminnallisuuden kehittämisessä on pidetty samalla silmällä sovelluksen mukautuvaisuutta sovelluksiin, joissa liikeprofiilit ovat aivan toisenlaisia. Pienempi taulukkoalkioiden määrä lyhentää ainakin nokkapyöräprofiilin lataukseen käytettävää aikaa.

Taulukossa 2 on myös muutamia kilpailijoita, jotka ovat toteuttaneet sovelluksen ohjelmoinnilla eli leikkaustietoja muutetaan suoraan ohjelmasta. Menetelmä ei ole helppokäyttöinen, sillä käyttäjän olisi ensin opeteltava ohjelma. Suorustietojen antaminen ACSM1-laitteeseen ohjelmoimalla ei ole hyvä ratkaisu, sillä ohjelmointi ei sopisi yhteen kohdan 2.4 kanssa. Siinä painotettiin laitteen yleiskäyttöisyyttä ja varottiin keskittämästä laitetta liikaa yhteen erikoissovellukseen.

Laitteeseen on jo toteutettu nokkapyörätoiminnallisuus, joten on hyvä, jos sitä voidaan käyttää lentävään leikkaukseen. Sen käyttöön liittyy kuitenkin ongelmia, joiden korjaaminen on osoittautunut hankalaksi. Synkronointisäädöstä johtuen yli kierroksen pituisten leikkauspalojen leikkaaminen onnistuu vain hyvin rajatuissa tapauksissa, profiilien luonti on hankalaa johtuen profiilien epäloogisesta paloittelusta usealle cam-taulukolle, nokkapyöräprofiilit ovat melko jäykkiä ja optimointi jää käyttäjän vastuulle. Edes aiemmin mainittu välityssuhteen käyttö nokkapyöräsovelluksessa ei poista näitä ongelmia.

Muutaman käyttöön ladatun nokkapyöräprofiilin vaihtaminen keskenään onnistuu graafisen työkalun toiminnallisuuksien avulla, ja profiilien välille voidaan luoda jonkinlaisia sekvenssejä. Kokonaan uusien profiilien lataaminen käyttöön vaatii lentävä

leikkaus -prosessin pysäyttämisen ja uudelleen lataamisen. Syy nokkapyörätoiminnallisuuden käyttöön lentävä leikkaus -sovelluksessa on kuitenkin vähentää tarvittavaa laskentatehoa. Profiilin pisteet on laskettu muutamalla kaavalla etukäteen taulukkoon, jolloin pisteiden laskeminen ei vie CPU-aikaa. Taulukon käyttöön tarvitaan enemmän muistia, mutta sitä saadaan teknologiakirjastosta. Lisäksi nokkapyörätoiminnallisuus on asiakasvaatimuksena joissakin tapauksissa.

Toinen vaihtoehto lentävän leikkauksen toteuttamiseen on tuottaa säätö parametrisoimalla, jolloin profiili laskee piirtämisen sijaan paikoituspisteitä parametritietojen pohjalta. Silloin pystytään käyttäjän puolesta paremmin optimoimaan energian ja ajan käyttöä profiilissa, lisäämään erilaisia toimintoja leikkauksen toteuttamiseen liittyen ja ennen kaikkea tuottamaan tavoitemittaisia leikkauspaloja. Puutteena voidaan pitää sitä, että säädön toteuttaminen on suunniteltava alusta ja tarvitaan enemmän CPU-aikaa.

Paikoituspisteiden määrittäminen toteutetaan ensin integroimalla loogisiin jaksoihin jaettua nopeusprofiilia, jossa jokaisella jaksolla on omat nopeuden lausekkeensa. Lausekkeita integroidaan jaksoon käytettävän ajan t_{period} suhteen, jossa period tarkoittaa käsiteltävää jaksoa. Integraalista saadaan ensin matka $s_{C,period}$, jonka kuljetin voi jakson aikana liikkua. Matka $s_{C,period}$ on vakio ja sitä tarvitaan paikoituskaavojen muodostamisessa myöhemmin. Paikoituksen toteuttamiseksi tarvitaan myös linjan paikan oloarvo s_L , jota verrataan linjan jakson aikana kulkemaan maksimaaliseen matkaan $s_{L,period}$. Muodostuva suhde kerrotaan jakson kestolla t_{period} :

$$t = \left(\frac{s_L}{s_{L,period}} \right) t_{period}. \quad (4.1)$$

Kun muodostuvaa ajan oloarvoa t käytetään kuljettimen nopeuden lausekkeen integroinnissa, saadaan kuljettimen paikalle oloarvo linjan paikan funktiona. Kuljettimen paikan oloarvo on käytännössä paikoituspiste, jonka informaatio syötetään kuljettimen paikoitussäätimeen paikoituksen toteuttamista varten. Uusi paikoituspiste lasketaan 500 μ s:n tai jopa 250 μ s:n välein riippuen kuinka usein linjan paikkatietoa luetaan. Leikkausprofiili saadaan, kun säädin interpoloi eli yhdistää paikoituspisteet. Paikoituspisteiden määrittäminen jatketaan, kunnes jakson tavoite on saavutettu. Sen jälkeen siirrytään integroimaan seuraavien jaksojen nopeuden laskentakaavoja, kunnes koko nopeusprofiili on käyty läpi (Voloshkin 2009).

Vaihtoehtona parametrisoinnin toteuttamiseen on edelleen jakaa leikkausprofiilin suorittaminen jaksoihin, mutta yhtäjaksoisesti suoritettavan interpoloinnin sijaan paikoituspisteet lasketaan etukäteen ja sijoitetaan omiin cam-tilaukkoihin. Valmiita cam-tilaukkoja ajetaan järjestyksessä leikkausprofiilin suorittamiseksi. Menetelmä vastaa käytännössä nokkapyörätoiminnallisuutta. Erona on, että tilaukkot ovat dynaamisia, profiileja ei tarvitse piirtää, tilaukkopisteitä tarvitaan vähemmän ja liikkeen optimointi saadaan suoritettua matemaattisilla malleilla. Ongelmaksi muodostuu kuitenkin nokkapyöräprofiilien muuttaminen kesken ajon ja kuljettimen palautus alkuasemaan. Siksi leikkausprofiilin paikoituspisteiden yhtäjaksoinen laskeminen nopeusprofiilin integroinnista ja paikoituspisteiden interpolointi on paras vaihtoehto toteuttaa sovellus.

Taulukko 2. Sovellustoimittajien tavat toteuttaa lentävä leikkaus -sovellus, ja lisäksi muutamia huomioita.

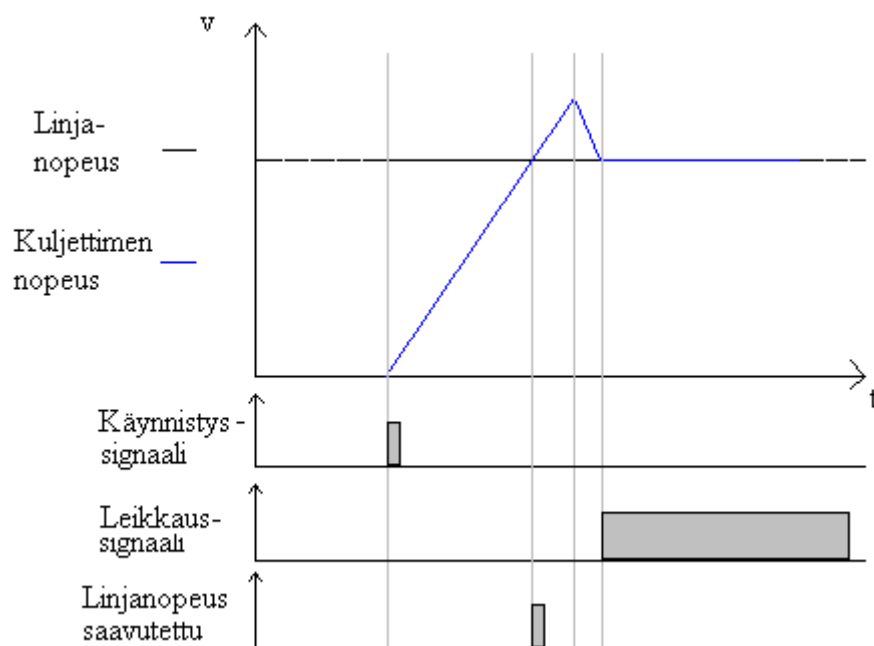
Sovellustoimittaja	Sovelluksen toteutus			Muuta
	Nokka-pyörä	Paramet-risoitava	Ohjelmoitu	
Lenze		X	(X)	Vakiototeutus parametri-soitava, ohjelmoitavissa erikseen asiakasvaatimusten mukaisiksi
CT	X	X		Parametritiedoista luodaan 10 pisteen cam-taulukko
Emerson	X	X		Käyttää CT:n järjestelmää
Rockwell			X	Pyörivä veitsi -sovellus tehty nokkapyörätoimin-nallisuudella
Yaskawa	X			Lentävä leikkuri ja pyörivä veitsi -sovellukset 1024 pisteisiin cam-taulukoihin
Siemens		X		Teknologiakirjastolla lisätään käyttöön moduuli, joka hoitaa leikkaussäädön toteutuksen joko välitys-suhteella, joka poistetaan käytöstä kuljettimen paluujakson aikana, tai lentävä leikkaus -mallilla, jossa säätö on jaettu perättäisiin jaksoihin. Jaksot suoritetaan tietyssä perättäisessä järjestyksessä. Parametrien asettelu on tehtävä joka kerta uudestaan syöttöjännitteen poiskytkemisen jälkeen.
SEW		X		Sovellus sijaitsee kirjastossa, joka ladataan erikseen käyttöön.
Baldor		X		
Soloist		X		
Parker		X		Lentävä leikkaus ja pyörivä veitsi -sovellukset parametrisoitavia.
Gefran		X		Sovellus integroitu osaksi käyttöä
Jetter		X		
Beckhoff			X	Sovellus ohjelmoidaan käyttäen PLC-kirjaston (Programmable logic controller) lohkoja
Ormec			X	Käytetään välityssuhdetta

4.2 Kuljettimen kiihdytys

Parametrisoinnissa lentävä leikkaus toteutetaan säätimessä jakamalla säätö loogisiin jaksoihin. Säätimen osien suoritusta voidaan haluttaessa vaihdella halutun lopputuloksen saamiseksi vain joitakin parametritietoja muuttamalla ilman käyrien piirtämistä ja niiden sovittelua. Seuraavana esitellään joidenkin toiminnallisuuksien vaikutukset leikkausprofiiliin.

Ylitahdistettu kiihdytys (Lenze 2008)

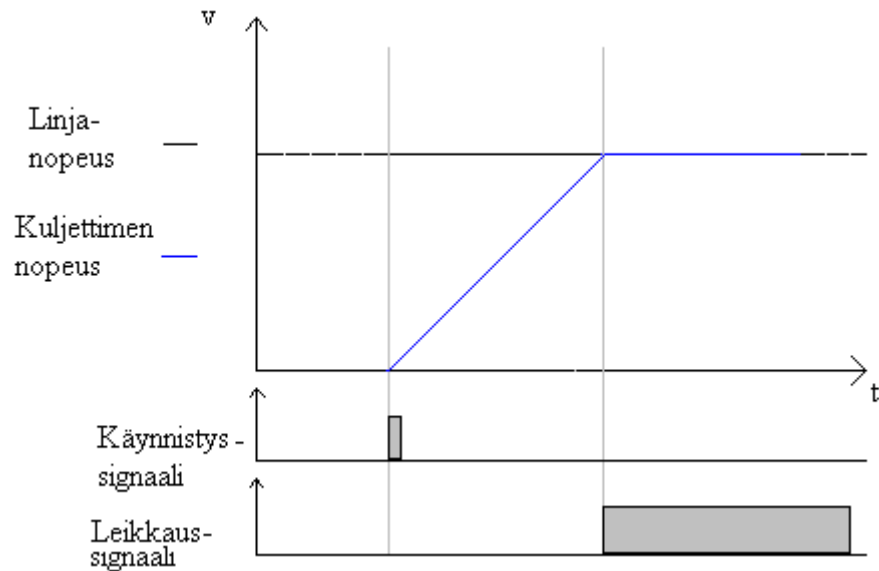
Kun leikkauksen käynnistävä signaali saadaan, leikkurin kuljetin tahdistetaan linjanopeuteen. Kuljettimen nopeus nostetaan hetkellisesti suuremmaksi kuin linjanopeus eli nopeus ylitahdistetaan, jotta leikkuri saadaan asettumaan myös oikeaan leikkauskohtaan. Nopeuksien tahdistuessa lähetetään ”linjanopeus saavutettu” -signaali ja paikan sekä nopeuden jälleen tahdistuessa lähetetään leikkaussignaali, jolloin leikkausoperaatio voi alkaa. Kuva 15 havainnollistaa asiaa.



Kuva 15. Ylitahdistetun kiihdytyksen profiili ja siihen läheisesti liittyvät signaalit.

Tahdistettu kiihdytys (Lenze 2008)

Tahdistetussa kiihdytyksessä kuljettimen nopeus ei koskaan ole suurempi kuin linjanopeus, kuten esitetty kuvassa 16. Tämä on mahdollista käyttämällä offset-toimintoa, jossa kuljetin lähtee liikkeelle ennen kuin materiaalin leikkauskohta saapuu kuljettimen alkuaseman kohdalle. Tällainen ominaisuus voi olla tarpeellinen silloin, kun materiaalin nopeus on hyvin korkea ja tahdistettava kuljetin ei voi saavuttaa oikeaa leikkauskohtaa ylittämättä nopeusrajaa. Tärkein syy käyttää tahdistettua kiihdytystä on energian käytön optimointi.

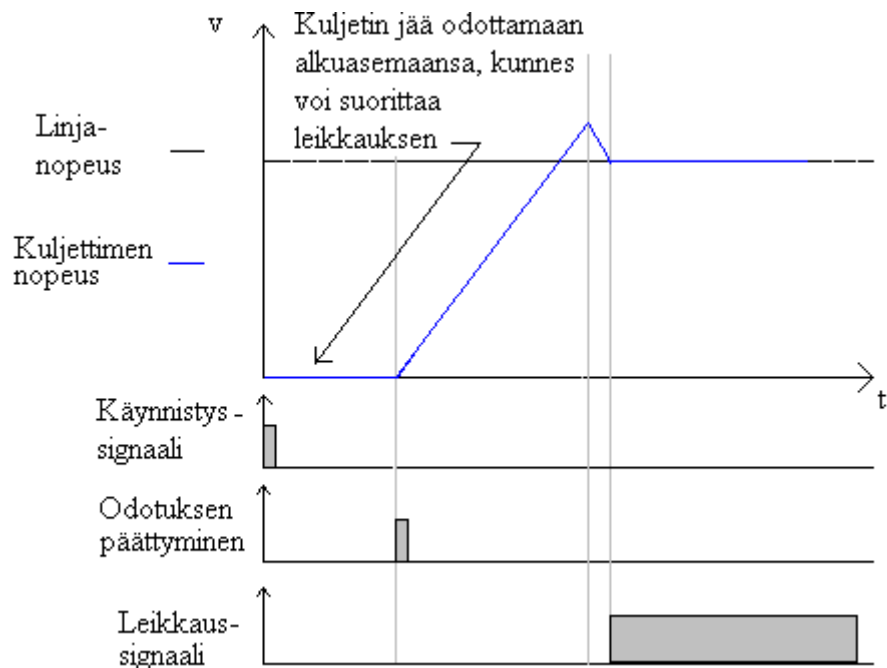


Kuva 16. Tahdistetun kiihdytyksen profiili ja siihen liittyvät signaalit.

Leikkauksen aloitus

Kun leikkaus aloitetaan, leikkausmateriaalin reuna voi olla huono. Tällöin kuljettimen on tarpeen odottaa alkuasemassa, että huono materiaali menee ensin ohi. Vaihtoehtona on määrittää leikkauksen aloituksen yhteyteen jokin aika, jonka kuljetin odottaa alkuasemassaan ennen leikkausprosessin aloitusta kuvan 17 tavalla. Aloituksen viipymisen jälkeen leikkaus alkaa normaalisti.

Toinen vaihtoehto huonon aloitusreunan käsittelyyn on tehdä se manuaalisesti eli antaa materiaalin vain mennä kuljettimen ohi. Käyttäjä määrittää itse materiaalia tarkkailemalla, milloin laatu on riittävän hyvä. Ohjelmaan ei tällöin erikseen tarvitse liittää mitään ylimääräistä toimintoa. Leikkaus aloitetaan lennosta leikkauksen käynnistävästä signaalista.

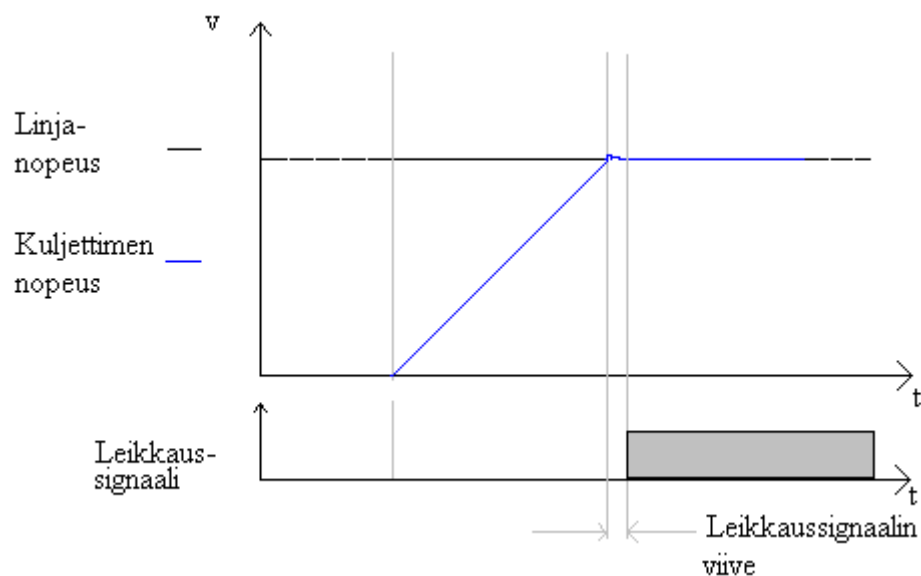


Kuva 17. Odotuksen vaikutus leikkausprofiilin nopeuskäyrään.

4.3 Leikkaus

Operoinnin viivästys

Kuljetin tahdistetaan oikeaan nopeuteen. Nopeus saattaa kuitenkin värähdellä jonkin aikaa, ennen kuin se asettuu lopulliseen arvoonsa (Lenze 2008). Siksi voi olla tarpeen asettaa viive leikkausprosessille kuvan 18 tapaan. Vaikka värähtelyä ei esiintyisikään, käyttäjä voi jostain syystä haluta viivästyttää operoinnin aloittamista. Viiveen suuruus määritellään millisekunteina, ja se lisätään myös suoraan leikkausjakson keston.



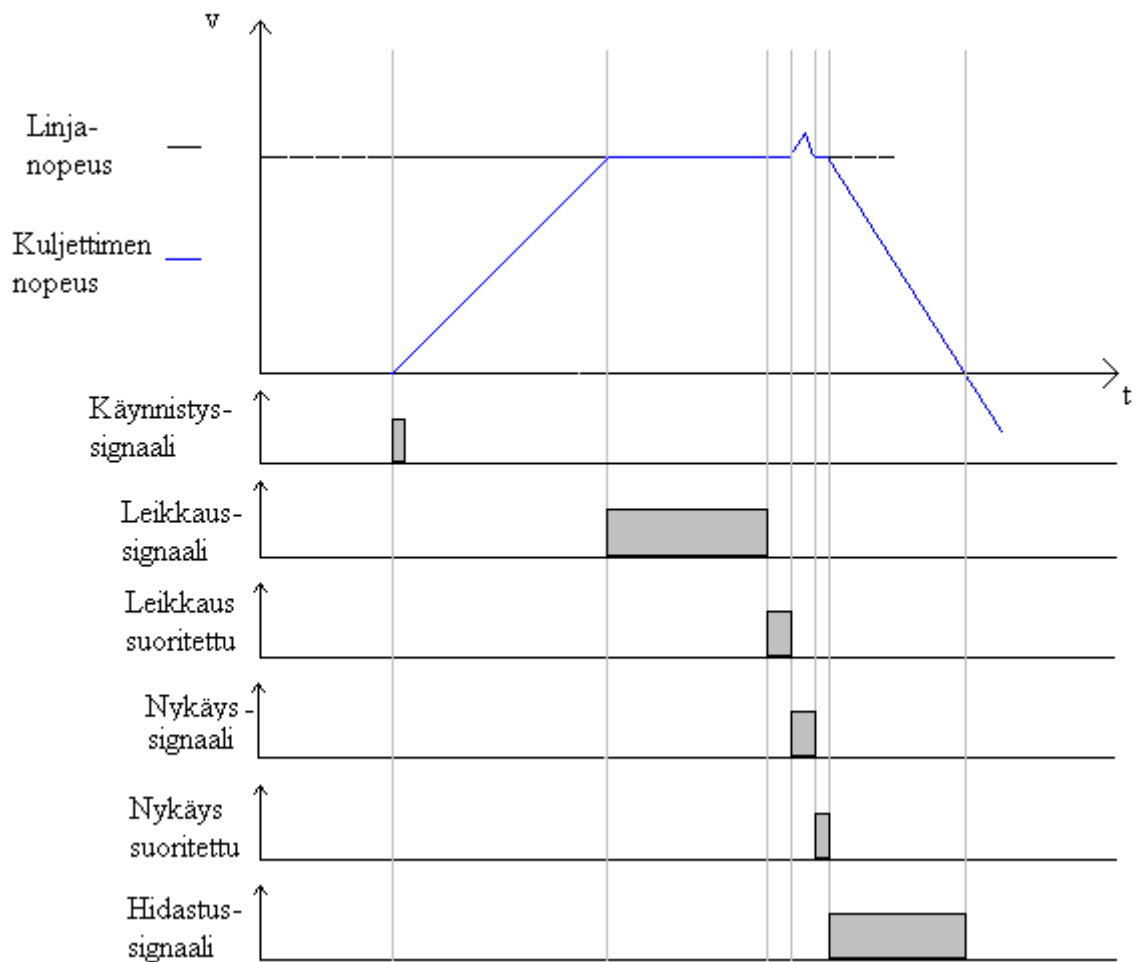
Kuva 18. Viivästetyn operoinnin periaatekuva.

Nykäys

Leikkauksen jälkeen voi olla tarpeen erottaa leikatut materiaalit toisistaan. Taajuusmuuttajasovelluksella leikatut materiaalit voidaan erotella leikkauksen jälkeen toisistaan nykäystoiminnolla. Toiminnossa leikkurin kuljetinta liikutetaan lyhyesti linjanopeutta suuremmalla nopeudella kuvan 19 tapaan (Lenze 2008).

Sahaussovelluksissa nykäystoiminto tulee kysymykseen vain, jos sahaustyökalu ei ole sahatun ja sahaamattoman osan välissä, sillä sahanterä turmelee helposti juuri leikatun reunan. Toisaalta, jos leikkausrakoa ei suurennetta ennen leikkausterän nostoa, terä saattaa jäädä jumiin eli raon suurentamista saatetaan tarvita jo aiemmin. Erottelu voidaan tehdä tällaisissa tapauksissa järjestelmäkäytöllä laajentamalla leikkausrakoa esimerkiksi mekaanishydraulisella laitteella, joka vetää sahatun osan sivuun (Varis 2009). Nykäyksen sopivuus on siis määritettävä tapauskohtaisesti, joten toiminnallisuuden on hyvä olla valinnainen.

Niin sahauksessa kuin leikkauksessakin nykäyksen suorittava signaali voidaan lähettää, kun saadaan signaali ”leikkaus suoritettu” ja on varmuus, että leikkausterät ovat ylhäällä. Nykäys voidaan suorittaa tietyssä ajassa, tai nykäyksen on saatava aikaan tietyn suuruinen väli leikkausreunojen välille (SEW 2005). Kun nykäys on suoritettu, lähetetään signaali nykäyksen suorittamisesta ja tämän jälkeen signaali hidastusvaiheen aloittamiseen.



Kuva 19. Nykäyksen suorittaminen leikkausjakson päätteeksi.

4.4 Leikkausprofiili

Tässä kohdassa määritetään eri jaksoissa tarvittavat paikoituspisteiden laskentakaavat. Niiden määrittämistä varten integroidaan nopeusprofiilia ajan suhteen. Sijoittamalla linjan paikan oloarvon ja ajan riippuvuus kaavasta (4.1) integraaliin, saadaan seuraava paikoituspiste. Paikoituspisteiden interpolointia suoritetaan aikatazon määrittämisen keston aikavälein, jolloin paikoituspisteiden laskentaa suoritetaan yhden leikkauksen suorittamiseksi leikkausprofiilin keston ja aikatazon keston osamäärän verran. Paluujaksossa osa paikoituskaavoista lasketaan paloittain, ja niiden määrittämiseen liittyy joitakin ehtoja. Tarkastellaan paluujaksoa sen takia omana kohtanaan.

Leikattava materiaali liikkuu oletusarvoisesti vakionopeudella v_L eteenpäin ja materiaalin leikkauspituus s_{length} saadaan parametritiedoista. Yhteen leikkausprofiiliin voidaan käyttää aika

$$t_{profile} = \frac{s_{length}}{v_L}. \quad (4.2)$$

Tänä aikana tehdään kiihdytys, paikoituksen korjaus, leikkaus, jarrutus sekä palautuminen takaisin alkuasemaan. Jakson lopussa on tauko, ja sille on mahdollista määrittää jokin kesto riippuen paluujakson tilasta.

Koko leikkausprofiiliin käytettävä aika voidaan jakaa jaksoihin:

$$t_{profile} = t_{acc} + t_{presync} + t_{sync} + t_{dec} + t_{ret} + t_{pending}, \quad (4.3)$$

missä

t_{acc} on kiihdytykseen käytetty aika,

$t_{presync}$ paikoitusvirheen korjaamiseen käytettävä aika,

t_{sync} leikkausoperaatioon käytetty aika,

t_{dec} kuljetin pysäytykseen käytetty aika,

t_{ret} kuljetin alkuasemaan siirtymisaika ja

$t_{pending}$ tauon pituus, jolloin kuljetin ei liiku.

Kuljetin liikkuu leikkausprofiilin aikana erisuuruisen matkan linjan kanssa. Leikkausjaksoissa kuljetin liikkuu radalla eteenpäin matkat $s_{C,acc}$, $s_{C,presync}$, $s_{C,sync}$ ja $s_{C,dec}$, jossa

$s_{C,acc}$ on kiihdytykseen käytetty matka,

$s_{C,presync}$ paikoitusvirheen korjaamisen aikana liikuttava matka,

$s_{C,sync}$ leikkauksen aikana kuljettu matka ja

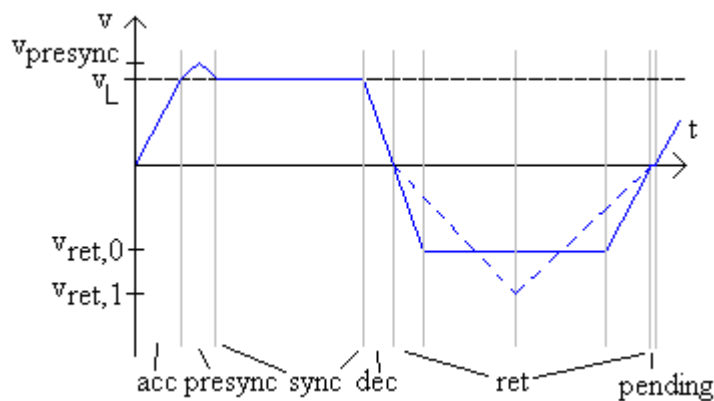
$s_{C,dec}$ kuljetin pysähtymiseen menevä matka.

Leikkausjaksojen jälkeen kuljetin liikkuu takaisin alkuasemaan matkan $s_{C,ret}$ eli matkan, joka tarvittiin leikkausjaksoihin:

$$s_{C,ret} = s_{C,acc} + s_{C,presync} + s_{C,sync} + s_{C,dec}. \quad (4.4)$$

Taukojakson aikana kuljetin ei liiku, joten $s_{C,pending} = 0$. Eri jaksojen vaikutus kuljetin nopeuteen on piirretty kuvaan 20.

Liikeprofiilin määrittämisessä on tavoitteena automatisoida liikeprofiilin luontia ja poistaa kiihtyvyydenmäärittämissparametrit eli pitää käyttäjältä saatavien leikkausparametrien määrä mahdollisimman alhaisena. Automaatio 2009 -messuilla 23.9.2009 kävi kuitenkin ilmi, että useimmat asiakkaat haluavat itse määrittää synkronoinnin aloituskohdan, joten kiihtyvyydenjakson ja paikoitusvirheen korjaus -jakson määrittämissparametrit on kuitenkin otettava käyttöön synkronoinnin aloituksen määrittämisen mahdollistamiseksi.



Kuva 20. Leikkausjaksojen, paluujakson ja tauon vaikutus kuljetin nopeuden käyrään.

Kiihdytysjaksossa kuljetin kiihdytetään oikeaan linjanopeuteen kuvan 21 lineaarisessa osassa. Kiihtymiseen tarvittava aika lasketaan

$$t_{acc} = \frac{v_L}{a_{acc}}, \quad (4.5)$$

kun kiihtyvyydestieto a_{acc} saadaan käyttäjältä parametritietona. Kiihdytysjakson päätteeksi kuljetettu matka saadaan integroimalla nopeutta ajan suhteen:

$$s_{C,acc} = \int_0^{t_{acc}} a_{acc} t dt = \frac{1}{2} a_{acc} t_{acc}^2 = \frac{v_L^2}{2a_{acc}}. \quad (4.6)$$

Linja on kiihdytysjaksoon käytettävän ajan t_{acc} aikana kulkenut matkan

$$s_{L,acc} = v_L t_{acc}. \quad (4.7)$$

Kiihdytysjaksossa kuljettimen paikka määritetään:

$$s_C = \int_0^t a_{acc} t dt = \int_0^t \frac{v_L}{t_{acc}} t dt = \frac{v_L}{t_{acc}} \frac{t^2}{2}. \quad (4.8)$$

Kaavassa (4.8) esiintyvä aika t saadaan kaavan (4.1) periaatteesta, jossa linjan paikan oloarvo s_L suhteutetaan linjan jakson aikana kulkemaan matkaan $s_{L,acc}$, ja kertomalla tämä jakson kestolla:

$$t = \frac{s_L}{s_{L,acc}} t_{acc}. \quad (4.9)$$

Sijoittamalla tämä kaavaan (4.8) kuljettimen paikan oloarvolle saadaan riippuvuus linjan paikasta:

$$s_C = v_L \frac{\left(\frac{s_L}{s_{L,acc}} \right)^2 t_{acc}}{2}. \quad (4.10)$$

Lopulta lausekkeita (4.5), (4.6) ja (4.10) käyttäen linjan paikan oloarvo voidaan esittää kuljettimen paikan oloarvon funktiona:

$$\boxed{s_{Ct} = s_{C,acc} \left(\frac{s_L}{s_{L,acc}} \right)^2}. \quad (4.11)$$

Kaavoja (4.6) ja (4.7) tarkastelemalla huomataan, että matkat ovat erisuuret. Se tarkoittaa sitä, että kuljetin ei ole saavuttanut vielä oikeaa leikkauskohtaa materiaalissa. Paikoitusvirheeksi muodostuu näin ollen

$$s_{error} = s_{L,acc} - s_{C,acc} = \frac{1}{2} v_L t_{acc} \quad (4.12)$$

eli kuljetin liikkuu kiihdytysjakson aikana täsmälleen puolet linjan kulkemasta matkasta.

Paikoitusvirheen korjaamiseksi suoritetaan paikoituksen korjaus -jakso, jossa kuljetinta liikutetaan linjanopeutta suuremmalla nopeudella paraabelin huipun muotoisella nopeuden käyrällä, jotta kuljetin saavuttaa oikean paikan eli leikkausmerkin linjalta. Kuljettimen on liikuttava tällöin linjan kulkema matka, mutta myös paikoitusvirheen suuruinen matka. Jaksossa käytettävää nopeuden käyrää on havainnollistettu kuvassa 21.

Monet käyttäjät haluavat itse määrittää leikkauksen aloituskohdan käynnistystehetä, joten kysytään matka $s_{cut,start}$ tai aika $t_{cut,start}$ käyttäjältä parametritietona. Matkanmääritysparametria käytettäessä leikkauksen aloitus tapahtuu ajanhetkellä

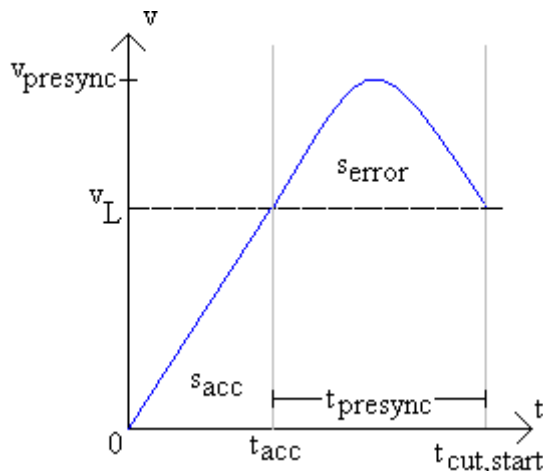
$$t_{cut,start} = \frac{s_{cut,start}}{v_L}. \quad (4.13)$$

Paikoitusvirheen kompensoimiseksi ja nopeuslausekkeen määrittämiseksi käytetään kuvaa 22. Kuljetinta liikutetaan jaksossa linjanopeutta suuremmalla nopeudella paraabelin huipun muotoista nopeuden liikerataa pitkin (Adams 2003). Paraabelin huippunopeus v_{max} on paraabelin puolivälissä, ja se määritetään ajanhetkelle $\frac{1}{2}t_{presync}$ paikoituksen korjaus -jakson kestäessä ajan

$$t_{presync} = t_{cut,start} - t_{acc}. \quad (4.14)$$

Paikoituksen korjaus -jakson virheenkorjauksesta aiheutuvan nopeuslisän lausekkeeksi muodostuu

$$v_{error}(t) = -4 \frac{v_{max}}{t_{presync}^2} t(t - t_{presync}). \quad (4.15)$$



Kuva 21. Kiihdytysjakson ja paikoituksen korjaus -jakson nopeusprofiili lähemmässä tarkastelussa.

Lausekkeessa (4.15) paraabelin huippunopeus on tuntematon, mutta se voidaan määrittää integroimalla lauseketta ajan $t_{presync}$. Tässä kohden on hyvä mainita, että jakson ajan laskenta t_{period} aloitetaan aina jakson alkaessa. Integraalista tulee paikoitusvirheen suuruinen matka

$$s_{error} = \int v_{error} dt = \int_0^{t_{presync}} -4 \frac{v_{max}}{2} \frac{t^2}{t_{presync}^2} + 4 \frac{v_{max}}{t_{presync}} t dt, \quad (4.16)$$

$$= -\frac{4}{3} v_{max} t_{presync} + 2 v_{max} t_{presync} = \frac{2}{3} v_{max} t_{presync}$$

joka tiedetään. Tätä apuna käyttäen saadaan paraabelin huippunopeus

$$v_{max} = \frac{3}{2} \frac{s_{error}}{t_{presync}}. \quad (4.17)$$

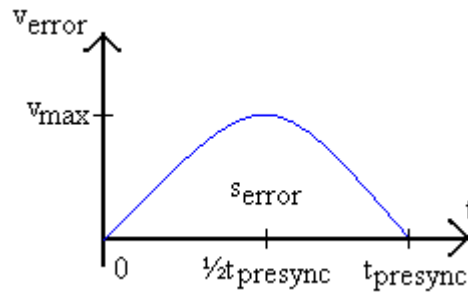
Paikoitusvirheen korjaus -jakson nopeus määritetään linjanopeuden ja nopeuslisän summana

$$v(t) = v_L(t) + v_{error}(t). \quad (4.18)$$

Koko jakson nopeuden lauseke voidaan esittää kaavoja (4.12), (4.15), (4.17) ja (4.18) hyväksi käyttäen muodossa

$$v(t) = v_L - 3 \frac{v_L t_{acc}}{t_{presync}^3} t(t - t_{presync}). \quad (4.19)$$

Nopeuden lauseke on voimassa kiihdytysjakson jälkeen ajan $t_{presync}$. Kuljetin on saavuttanut tällöin oikean paikan linjalta, ja se liikkuu linjanopeudella v_L .



Kuva 22. Paikoitusvirheen eliminoimiseen tarvittava nopeuden käyrä.

Kuljettimen kulkema matka paikoituksen korjaus -jaksossa voidaan laskea integroimalla nopeutta:

$$s_{C,presync} = \int_0^{t_{presync}} v_L - 3 \frac{v_L t_{acc}}{t_{presync}^3} t(t - t_{presync}) dt = v_L \left(t_{presync} + \frac{1}{2} t_{acc} \right). \quad (4.20)$$

Lausekkeessa $v_L t_{presync}$ on linjan kulkema matka $s_{L,presync}$ ja $\frac{1}{2} v_L t_{acc}$ paikoitusvirheen suuruinen matka s_{error} kaavasta (4.12).

Paikoituksen korjaus -jaksossa kuljettimen paikan oloarvo määritetään integroimalla nopeuden lauseketta (4.19) ajan t suhteen:

$$s_C = \int_0^t v_L - 3 \frac{v_L t_{acc}}{t_{presync}^3} t(t - t_{presync}) dt = v_L t \left[1 - \frac{t_{acc} t}{t_{presync}^2} \left(\frac{t}{t_{presync}} - \frac{3}{2} \right) \right], \quad (4.21)$$

jossa aika

$$t = \frac{s_L}{s_{L,presync}} t_{presync}. \quad (4.22)$$

Kuljettimen paikan oloarvolle ja linjan paikan oloarvolle saadaan näin ollen riippuvuus

$$\begin{aligned} s_C &= t_{presync} v_L \left(\frac{s_L}{s_{L,presync}} \right) \left\{ 1 - \frac{t_{acc} \left(\frac{s_L}{s_{L,presync}} \right)}{t_{presync}} \left[\left(\frac{s_L}{s_{L,presync}} \right) - \frac{3}{2} \right] \right\} \\ &= \left(\frac{s_L}{s_{L,presync}} \right) \left\{ s_{L,presync} - s_{L,acc} \left(\frac{s_L}{s_{L,presync}} \right) \left[\left(\frac{s_L}{s_{L,presync}} \right) - \frac{3}{2} \right] \right\} \end{aligned} \quad (4.23)$$

Lopulliseksi kuljettimen ja linjan paikan riippuvuudeksi paikoituksen korjaus -jaksossa tulee:

$$s_C = s_L - s_{L,acc} \left(\frac{s_L}{s_{L,presync}} \right)^2 \left[\left(\frac{s_L}{s_{L,presync}} \right) - \frac{3}{2} \right]. \quad (4.24)$$

Kun kuljetin saavuttaa oikean kohdan linjalta ja liikkuu linjan nopeudella, alkaa leikkausjakso. Leikkausjakson leikkausoperaatioon käytettävä aika t_{sync} kysytään käyttäjältä parametritietona tai haetaan kenttäväylää pitkin sahausohjelmistoilta. Leikkauksen kestoksi t_{sync} tulee asettaa niin suuri arvo, että leikkaus saadaan varmasti suoritettua, ja leikkaustyökalut ehtivät nousta ylös. Samalla tulee kuitenkin varoa, ettei leikkausjaksosta tehdä turhan pitkää.

Kuljetin ja linja liikkuvat leikkausjakson loppuun mennessä matkan

$$s_{C, sync} = s_{L, sync} = v_L t_{sync}. \quad (4.25)$$

Kuljettimen on seurattava orjallisesti linjan paikkaa, joten kuljettimen paikan oloarvon riippuvuus linjan paikasta leikkausjaksossa voidaan esittää kaavan (4.1) nojalla muodossa

$$\boxed{s_C = s_{C, sync} \left(\frac{s_L}{s_{L, sync}} \right)}, \quad (4.26)$$

missä s_L tarkoittaa linjan oloarvoa leikkausjaksossa.

Leikkauksen suorittamisen jälkeen kuljetin suorittaa hidastusjakson, jolloin kuljetin hidastuu ja pysähtyy. Kiihtyvyys a_{dec} saadaan käyttäjältä parametritietona, joten kuljettimen pysähtymiseen käytetään jarrutusjaksossa aika

$$t_{dec} = \left| -\frac{v_L}{a_{dec}} \right| = \frac{v_L}{a_{dec}}. \quad (4.27)$$

Jarrutusjaksossa integroidaan seuraavaa nopeuden lauseketta jakson aikana kuljettavan matkan määrittämiseksi:

$$s_{C, dec} = \int_0^{t_{dec}} (v_L - a_{dec} t) dt = v_L t_{dec} - \frac{1}{2} a_{dec} t_{dec}^2 = \frac{1}{2} \frac{v_L^2}{a_{dec}}. \quad (4.28)$$

Linja on liikkunut samassa ajassa itseisarvon matkasta

$$s_{L, dec} = v_L t_{dec}. \quad (4.29)$$

Jarrutusjaksossa kuljettimen paikan oloarvo määritellään lausekkeen (4.27) avulla seuraavasti:

$$s_C = \int_0^t (v_L - a_{dec} t) dt = v_L t - \frac{1}{2} a_{dec} t^2. \quad (4.30)$$

Koska linjan liike on lineaarista, on mahdollista määrittää suoraan riippuvuus linjan paikalle ja ajalle liikkeessä:

$$t = \frac{s_L}{s_{L, dec}} t_{dec}. \quad (4.31)$$

Sijoitetaan tämä lausekkeeseen (4.30) käyttäen hyväksi kaavaa (4.28):

$$\begin{aligned}
s_C &= v_L \left(\frac{s_L}{s_{L,dec}} t_{dec} \right) - \frac{1}{2} a_{dec} \left(\frac{s_L}{s_{L,dec}} t_{dec} \right)^2 \\
&= -2 \frac{1}{2} v_L \frac{v_L}{a_{dec}} \left(\frac{s_L}{s_{L,dec}} \right) + \frac{1}{2} a_{dec} \left(\frac{v_L}{a_{dec}} \right)^2 \left(\frac{s_L}{s_{L,dec}} \right)^2.
\end{aligned} \tag{4.32}$$

Lopulliseksi kuljettimen paikan ja linjan paikan väliseksi riippuvuudeksi saadaan

$$\boxed{s_C = s_{C,dec} \left(\frac{s_L}{s_{L,dec}} \right) \left[2 - \left(\frac{s_L}{s_{L,dec}} \right) \right]}. \tag{4.33}$$

Taukojaksossa kuljetin ei liiku, joten taukojaksossa paikan oloarvo määritellään

$$\boxed{s_C = 0}. \tag{4.34}$$

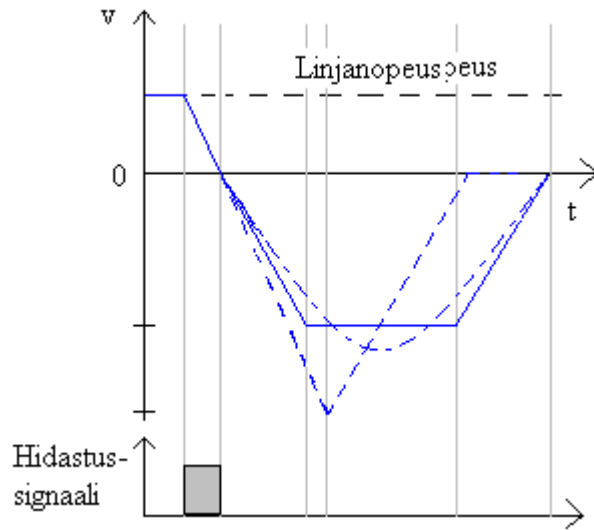
Paluujaksoon käytettävä aika ratkaistaan kaavalla

$$t_{ret} = t_{profile} - t_{acc} - t_{presync} - t_{sync} - t_{dec} - t_{pending}. \tag{4.35}$$

Paluujaksoon käytetään sen verran aikaa kuin muilta jaksoilta jää käyttämättä. Jos materiaalin leikkaukseen ja kiihdytyksiin käytetään liian paljon aikaa, kuljettimen absoluuttinen nopeus voidaan joutua nostamaan korkeaksi ja käyttämään suuria kiihtyvyyksiä niin, että nopeuden ja kiihtyvyyksien absoluuttiset maksimirajat tulevat vastaan. Siksi on syytä tarkistaa jo profiilin luonnin yhteydessä, että kuljettimen liikkeen toteuttaminen onnistuu lasketuilla arvoilla. Lisäksi on huomioitava, että paluujakson kriittiset parametrit saattavat muuttua leikkauksen aikana, jos linjanopeus muuttuu huomattavasti.

4.5 Paluuvaiheen profiilit

Paluujakso voidaan suorittaa kolmella tavalla. Kuvassa 23 on esitetty kaikkien niiden paluunopeuksien kuvaajat. Katkoviivoitetun kolmion kuvaaja esittää aikaoptimointia, jossa paluuvaihe suoritetaan mahdollisimman nopeasti rajoittamatta absoluuttista maksiminopeutta. Puolisuunnikasta esittävä kuva tarkoittaa energiaoptimointia, jolloin maksiminopeutta rajoitetaan ja palautus suoritetaan juuri ajoissa ennen seuraavaa leikkausta. Paraabelin muotoinen kuvaaja esittää toteutuksen optimointia. Tätä profiilimallia voidaan käyttää hyödyksi sekä energia- ja aikaoptimoinnissa.



Kuva 23. Aikaoptimoidun, energiaoptimoidun ja toteutuksen optimoinnin paluuvaiheen nopeuksien pelkistetyt profiilit.

4.5.1 Energiaoptimoitu paluuvaihe

Jos leikkaus suoritetaan määrätyn pituuden mukaan, ehditään suorittamaan profiililaskenta ennen seuraavaa leikkausta. Kuljetin optimoidaan alkuasemaansa juuri ennen seuraavan profiilin alkamista, jolloin voidaan optimoida energian käyttöä välttämällä turhat kiihdyttelyt ja suuret nopeudet, jolloin kyseessä on energiaoptimointi.

Energiaoptimoitu paluuvaihe suoritetaan puolisuunnikkaan muotoisella profiililla. Määrityksen apuna käytetään kiihdytyksestä ja jarrutuksesta koostuvaa vertojaksoa sekä kiihdytyksestä, vakionopeudesta ja jarrutuksesta koostuvaa vakionopeusjaksoa kuvan 24 tapaan. Vertojaksen vertonopeudelle $v_{ret,1}$ ajaminen vie enemmän energiaa kuin vakionopeusjaksossa vakionopeudella $v_{ret,0}$ ajaminen. Pienemmillä nopeuksilla puolestaan tarvitaan suurempia kiihtyvyyksiä, jolloin tarvitaan suurempi momentti. On siksi valittava sopiva suhde vertojaksen ja vakionopeusjakson nopeuksien välille.

Vertonopeuteen kiihdyttäminen kestää puolet paluujakson kestosta eli $t_{ret,1} = \frac{1}{2}t_{ret}$. Kuljetin liikkuu tällöin matkan $s_{C,ret,1}$, joka puolet paluujakson matkasta $s_{C,ret}$. Vertonopeuteen kiihdyttäminen voidaan tällöin määrittää kaavalla

$$a_{ret,1} = \frac{v_{ret,1}}{t_{ret,1}} = \frac{s_{C,ret}}{t_{ret}^2}. \quad (4.36)$$

Vastaavasti vakionopeuteen kiihdyttäminen vie ajan $t_{ret,0}$, joten vakionopeusjakson kiihtyvyys on

$$a_{ret,0} = \frac{v_{ret,0}}{t_{ret,0}}. \quad (4.37)$$

Vakionopeuden ja vertonopeuden välillä on suhde

$$\xi = \frac{v_{ret,0}}{v_{ret,1}}, \quad (4.38)$$

jossa ξ voi saada arvoja väliltä 0 ja 1 (Luomi 2008).

Paluujakson puoliväliin $t_{ret,1}$ mennessä vertojaksossa on liikuttu puoleen väliin paluujaksoa eli

$$\frac{s_{C,ret}}{2} = \frac{1}{2} a_{ret,1} t_{ret,1}^2 = \frac{v_{ret,1}^2}{2a_{ret,1}}. \quad (4.39)$$

Paluujakson puoliväliin $t_{ret,1}$ mennessä vakionopeusjakson aikana kuljetaan myös matka

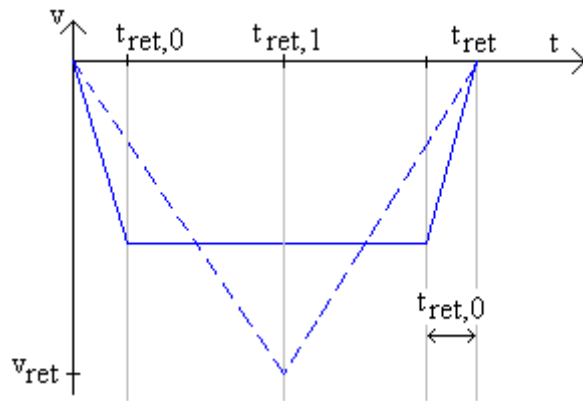
$$\begin{aligned} \frac{s_{C,ret}}{2} &= \frac{1}{2} t_{ret,0} \xi v_{ret,1} + (t_{ret,1} - t_{ret,0}) \xi v_{ret,1} = \left(t_{ret,1} - \frac{1}{2} t_{ret,0} \right) \xi v_{ret,1} \\ &= \left(\frac{v_{ret,1}}{a_{ret,1}} - \frac{1}{2} \frac{\xi v_{ret,1}}{a_{ret,0}} \right) \xi v_{ret,1} = \left(\frac{1}{a_{ret,1}} - \frac{1}{2} \frac{\xi}{a_{ret,0}} \right) \xi v_{ret,1}^2. \end{aligned} \quad (4.40)$$

Kuljetut matkat ovat yhtä suuria, joten kiihtyvyyksien suhteeksi saadaan tällöin

$$a_{ret,0} = \frac{\xi^2}{2\xi - 1} a_{ret,1}. \quad (4.41)$$

Vakionopeuteen kiihdyttämiseen tarvittava aika määritetään kaavasta (4.37), (4.38) ja (4.41). Vakionopeuden kiihdytysajaksi saadaan

$$t_{ret,0} = \frac{\xi v_{ret,1}}{\frac{\xi^2}{2\xi - 1} a_{ret,1}} = \frac{2\xi - 1}{\xi} t_{ret,1}. \quad (4.42)$$



Kuva 24. Paluujakson vakio- ja vertonopeuden profiilit. Pinta-alat ovat yhtä suuret.

Paluujakson suurin mahdollinen absoluuttinen nopeus on vertojakson nopeus $v_{ret,1}$, ja se saadaan määritettyä kolmion huipusta. Vakionopeusjaksosta saadaan puolestaan paluujakson pienin mahdollinen absoluuttinen paluunopeus kiihtyvyyksien lähestyessä ääretöntä. Kuvan 24 puolisuunnikas alkaa tällöin muistuttaa nelikulmiota. Kuljetut matkat ovat ääritapauksissa yhtä suuret, joten vakionopeuden ja vertonopeuden suhteen on oltava vähintään $\xi > 0,5$.

Kuljettimen kiihdytys vakionopeuteen

Energiaoptimoidun paluujakson paikoituksessa hyödynnetään kuvan 24 kolmeen osaan jaettua nopeusprofiilia. Paikoituksessa jokaiselle osalle on määritettävä omat riippuvuutensa linjan paikasta. Ensimmäisessä osassa, jossa kuljetin kiihdytetään vakionopeuteen, nopeus määritetään kaavalla

$$v(t) = -a_{ret,0}t. \quad (4.43)$$

Nopeus integroidaan ajan suhteen, jotta saadaan kuljettimen kulkema matka vakionopeuteen $v_{ret,0}$ kiihdytettäessä. Kiihdyttäminen kestää ajan $t_{ret,0}$, joten kuljettimen kulkema matka paluujakson vakionopeuteen kiihdytettäessä on

$$s_{C,ret,0} = \int_0^{t_{ret,0}} v(t)dt = \int_0^{t_{ret,0}} -a_{ret,0}t dt = -\frac{1}{2}a_{ret,0}t_{ret,0}^2. \quad (4.44)$$

Linja kulkema matka vakionopeudelle kiihdytettäessä on vastaavasti

$$s_{L,ret,0} = v_L t_{ret,0}. \quad (4.45)$$

Matkat ovat vastakkaismerkkiset, sillä kuljetin ja linja liikkuvat vastakkaisiin suuntiin.

Kuljettimen paikalle ja linjan paikalle saadaan riippuvuus paluujaksossa relaatiolla:

$$t = \frac{s_L}{s_{L,ret,0}} t_{ret,0}. \quad (4.46)$$

Kun paikan oloarvo vakionopeuteen kiihdytyksessä määritetään integroimalla nopeuden kaavaa (4.43) ajan t suhteen, saadaan kuljettimen paikan oloarvoksi

$$s_C = \int_0^t -a_{ret,0}t dt = \frac{1}{2} \frac{v_{ret,0}}{t_{ret,0}} \left(\frac{s_L}{s_{L,ret,0}} t_{ret,0} \right)^2 = \frac{1}{2} v_{ret,0} t_{ret,0} \left(\frac{s_L}{s_{L,ret,0}} \right)^2, \quad (4.47)$$

jossa paluunopeus $v_{ret,0}$ on negatiivinen. Käyttämällä kaavaa (4.37) ja (4.41) saadaan kuljettimen paikan oloarvo muotoon

$$s_C = \frac{1}{2} \frac{v_{ret,0}^2}{a_{ret,0}} \left(\frac{s_L}{s_{L,ret,0}} \right)^2 = \frac{1}{2} \frac{\xi^2 v_{ret,1}^2}{\frac{\xi^2}{2\xi-1} a_{ret,1}} \left(\frac{s_L}{s_{L,ret,0}} \right)^2. \quad (4.48)$$

Lopulliseksi paikan riippuvuudeksi paluujakson vakionopeusalueella tulee kaavaa (4.39) käyttäen

$$s_C = -\frac{(2\xi-1)}{2} \left| s_{C,ret} \right| \left(\frac{s_L}{s_{L,ret,0}} \right)^2. \quad (4.49)$$

Kuljetin liikkuu paluujaksossa päinvastaiseen suuntaan kuin linja, joten kuljettimen ja linjan paikan välillä on tällöin negatiivinen riippuvuus. Kaavassa on myös nopeussuhteen ξ vaikutus mukana. Sille selviää sopivat arvot myöhemmin tässä kohdassa 4.5.1.

Kuljettimen liikkuminen vakionopeudella

Kuvan 24 toisessa osassa kuljetin liikkuu vakionopeudella päinvastaiseen suuntaan linjaan nähden. Nopeusprofiilin nopeus määritetään kaavalla

$$v(t) = v_{ret,0}. \quad (4.50)$$

Kaavaa käytetään vakionopeuden saavuttamisen jälkeen ajan $t_{ret} - 2t_{ret,0}$. Kuljetin liikkuu tässä ajassa matkan

$$s_{C,ret,0} = \int_0^{t_{ret}-2t_{ret,0}} v_{ret,0} dt = 2\xi v_{ret,1} \left(\frac{v_{ret,1}}{a_{ret,1}} - \frac{\xi v_{ret,1}}{\frac{\xi^2}{2\xi-1} a_{ret,1}} \right) = 2(1-\xi) s_{C,ret}, \quad (4.51)$$

jossa ajan määrittämisessä käytetään kaavoja (4.36), (4.37), (4.38), (4.39) ja (4.41). Linja liikkuu samassa ajassa edellisen kaavan päättelyillä matkan

$$s_{L,ret,0} = v_L (t_{ret} - 2t_{ret,0}) = 2v_L t_{ret,1} \frac{1-\xi}{\xi}. \quad (4.52)$$

Kuljettimen paikan oloarvolle s_C ja linjan paikalle saadaan riippuvuus paluujakson vakionopeusalueella käyttämällä relaatiota:

$$t = \left(\frac{s_L}{s_{L,ret,0}} \right) (t_{ret} - 2t_{ret,0}). \quad (4.53)$$

Paikan oloarvo vakionopeudella liikuttaessa saadaan integroimalla kaavan (4.50) nopeutta:

$$s_C = \int_0^t v_{ret,0} dt = 2\xi v_{ret,1} \left(\frac{v_{ret,1}}{a_{ret,1}} - \frac{\xi v_{ret,1}}{\frac{\xi^2}{2\xi-1} a_{ret,1}} \right) \left(\frac{s_L}{s_{L,ret,0}} \right). \quad (4.54)$$

Kuljettimen ja linjan paikan väliseksi riippuvuudeksi tulee viimein

$$s_C = -2(1-\xi) s_{C,ret} \left(\frac{s_L}{s_{L,ret,0}} \right). \quad (4.55)$$

Kuljettimen pysäytys

Viimeinen osa energiaoptimoidun paluunopeuden profiilista määritetään lausekkeella

$$v(t) = v_{ret,0} + a_{ret,0}t, \quad (4.56)$$

ja se on voimassa, kunnes kuljetin pysähtyy alkuasemaansa. Kuljetin kulkee tällöin saman matkan kuin paluujakson vakionopeuteen kiihdytettäessä, joten käytetään paluujakson vakionopeuteen kiihdyttämisen tietoja hyväksi paikan oloarvon määrittämisessä. Integroimalla nopeutta, saadaan kuljettimen ja linjan paikan oloarvojen riippuvuudeksi

$$\begin{aligned} s_C &= \int_0^t -|v_{ret,0}| + a_{ret,0}t dt = -v_{ret,0} \left(\frac{s_L}{s_{L,ret,0}} t_{ret,0} \right) + \frac{1}{2} a_{ret,1} \left(\frac{s_L}{s_{L,ret,0}} t_{ret,0} \right)^2 \\ &= \left(\frac{s_L}{s_{L,ret,0}} \right) \left[(1 - 2\xi) \frac{v_{ret,1}^2}{a_{ret,1}} + \frac{1}{2} (2\xi - 1)^2 \frac{(v_{ret,1})^2}{a_{ret,1}} \left(\frac{s_L}{s_{L,ret,0}} \right) \right] \end{aligned} \quad (4.57)$$

Lopulta paluujakson absoluuttisesta huippunopeudesta nolla-asemaan hidastamiselle ja linjan paikalle muodostuu riippuvuus:

$$s_C = - \left| s_{C,ret} \left(\frac{s_L}{s_{L,ret,0}} \right) \left[(1 - 2\xi) + \frac{1}{2} (2\xi - 1)^2 \left(\frac{s_L}{s_{L,ret,0}} \right) \right] \right|. \quad (4.58)$$

Energiaoptimointi

Energiaoptimoinnissa paluujakso halutaan suorittaa mahdollisimman pienellä energialla. Tässä luvussa on käsitelty tähän asti lineaarista liikettä, mutta sähkökäytöissä on usein kyse pyörimisliikkeestä. Liike on kuitenkin suoraan verrannollista pyörimisliikkeeseen, joten käytetään seuraavissa vääntömomentin, tehon ja energian tarkasteluissa kulmanopeutta.

Pyörimisliikkeen tapauksessa J on hitausmomentti, T on sähkömoottorin synnyttämä vääntömomentti kulmanopeuden ω suunnassa ja T_L on liikettä vastustava kuormitusvääntömomentti. Liiketytälöksi saadaan yleisessä tapauksessa

$$T_a = T - T_L = J \frac{d\omega}{dt} + \omega \frac{dJ}{dt}, \quad (4.59)$$

missä $J\omega$ on liikemäärä ja T_a kiihdyttävä momentti (Luomi 2008). Yleensä hitausmomentti on vakio, joten yhtälö (4.59) yksinkertaistuu tavalliseksi liikeytälöksi

$$T = T_L + J \frac{d\omega}{dt}. \quad (4.60)$$

Käytännössä on aina mukana kuormituksen aiheuttamaa vastamomenttia ja energian siirron häviöitä. Oletetaan vastamomentti $T_L = 0$. Lisäksi liike-energiaa ei palauteta, vaan se tuhletaan häviöiksi jarruvastuksessa. Koska analysointi suoritetaan suhteellistarkasteluna, vääntömomenttien suhde kiihdytyksessä esitetään muodossa

$$\frac{T_{ret,0}}{T_{ret,1}} = \frac{a_{ret,0}}{a_{ret,1}} = \frac{\xi^2}{2\xi - 1}. \quad (4.61)$$

Kuvaan 25 on piirretty suhteellisen kiihtyvyyden riippuvuus nopeussuhteesta ξ . Huomataan, että huippunopeuden pienentäminen vaikuttaa aluksi hyvin vähän momentin suhteellisarvoon, joka on verrannollinen kiihtyvyyteen. Kasvu kuitenkin voimistuu voimakkaasti suhteen pienetessä. Raja-arvona voidaan pitää nopeussuhdetta 0,75, jolloin vakionopeuteen tarvittava kiihtyvyys on kasvanut 12,5 %. Tämän jälkeen tarvittava kiihtyvyys kasvaa voimakkaasti.

Kertomalla yhtälö (4.60) kulmanopeudella, saadaan tehotasapainoksi

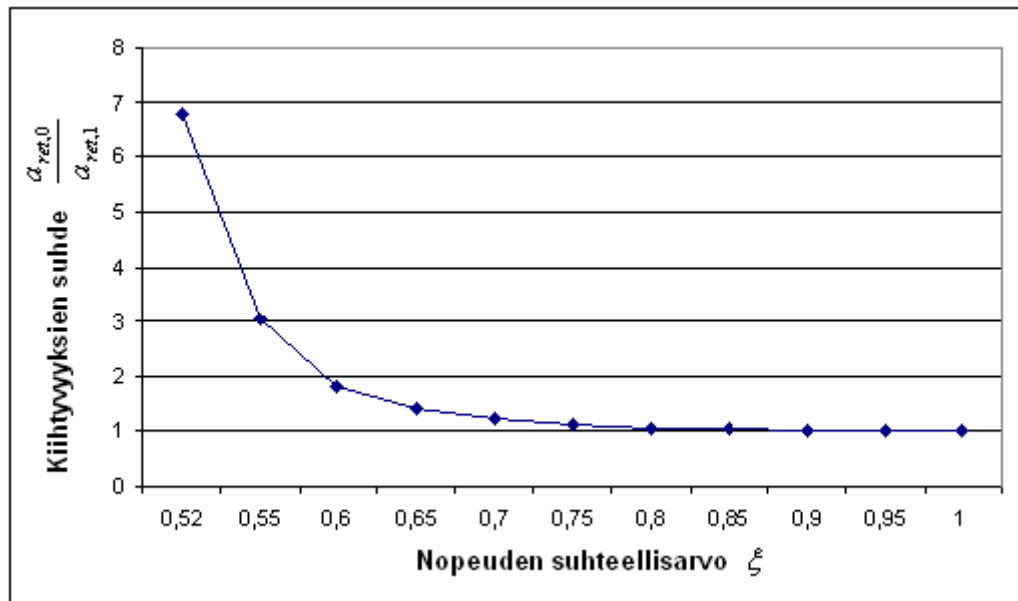
$$P = \omega T = \omega T_L + J\omega \frac{d\omega}{dt}, \quad (4.62)$$

missä $P_L = \omega T_L$ on kuormitusteho ja loput termit kuvaavat pyörivän massan liike-energian muutosta (Luomi 2008). Yhtälö yksinkertaistuu muotoon

$$P = J\omega \frac{d\omega}{dt}, \quad (4.63)$$

kun J on vakio ja $T_L = 0$. Kaavaa (4.63) hyödyntäen, tehon suhteelliset maksimiarvot voidaan esittää paluujaksossa muodossa

$$\frac{P_{ret,0}}{P_{ret,1}} = \frac{v_{ret,0}}{v_{ret,1}} \frac{T_{ret,0}}{T_{ret,1}} = \frac{\xi v_{ret,1}}{v_{ret,1}} \frac{a_{ret,0}}{a_{ret,1}} = \xi \frac{\xi^2}{2\xi - 1} = \frac{\xi^3}{2\xi - 1}. \quad (4.64)$$



Kuva 25. Suhteellisen kiihtyvyyden riippuvuus nopeussuhteesta.

Kuvaan 26 on piirretty momentin ja tehon suhteet nopeussuhteen funktiona. Muodostuneiden käyrien yksittäiset pisteet tarkoittavat tehon ja momentin suhteellista suuruutta siinä pisteessä, kun vakionopeus on saavutettu eri nopeussuhteen ξ arvoilla. Suhteellinen tehon tarve pienenee aluksi nopeammin kuin suhteellisen momentti kasvaa, kun nopeussuhdetta alennetaan. Momentti alkaa nousta merkittävästi vasta, kun nopeussuhde ξ alittaa arvon 0,75. Tehon tarve on alentuneena aina arvoon 0,65 asti, jonka jälkeen tehontarve alkaa jälleen voimakkaasti nousta.

Integroimalla teho (4.63) ajan suhteen, saadaan paluujakson kiihdyttämiseen tarvittava energia

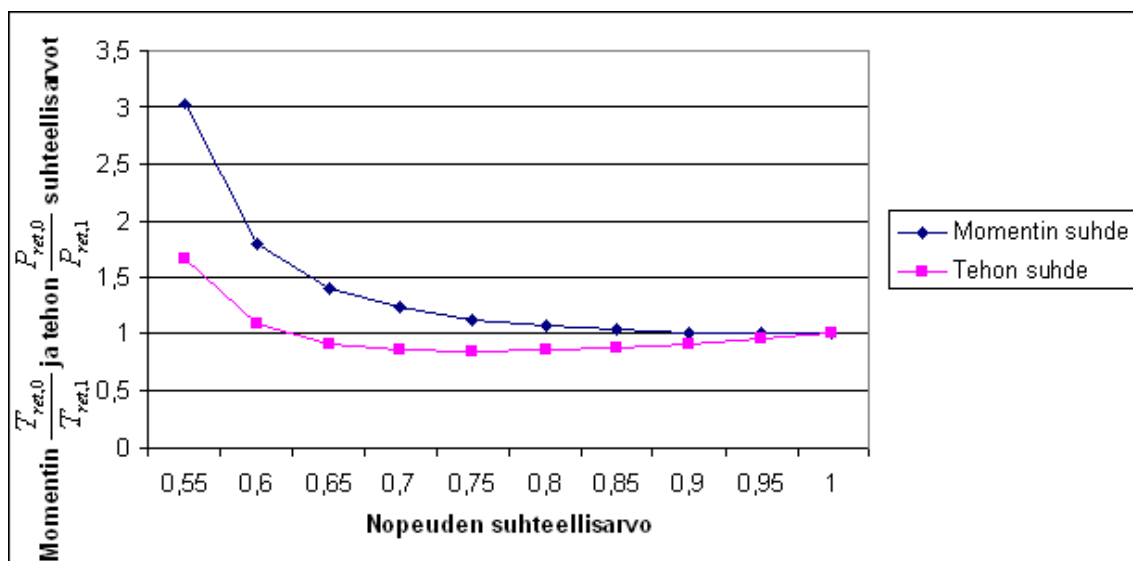
$$W(t) = \int_0^t P dt = \int_0^t J \omega \frac{d\omega}{dt} dt = \frac{1}{2} J \omega^2. \quad (4.65)$$

Kiihtyvyysjakson energia voidaan laskea kaavalla (4.65), sillä kulmanopeus muuttuu.

Vakionopeusjaksossa kulmanopeuden muutos $\frac{d\omega}{dt} = 0$, joten vakionopeusjaksossa kokonaisenergian kulutusta ei kasvateta. Suhteuttamalla vakionopeusjakson maksimienergia vertojakson energiaan, energian kulutukseen suhteeksi muodostuu

$$W(t) = \xi^2. \quad (4.66)$$

Energian määrittämisessä ei siis analysoida tehoa eikä momenttia, sillä energiankulutus riippu vain kulmanopeuden neliöstä. Energian kulutuksen kannalta nopeussuhde ξ kannattaa valita pienimmäksi mahdolliseksi eli lähelle $\xi \approx 0,5$. Pienimmällä mahdollisella nopeussuhteella muut mitoitus tekijät ovat kuitenkin suuria. Käytännössä nopeussuhde kannattaa valita suuremmaksi, jotta nämä muut mitoitus tekijät saadaan kohtuullisiksi.



Kuva 26. Momentin ja tehon suhteiden riippuvuus nopeuden suhteesta paluujakson kiihdytyksissä.

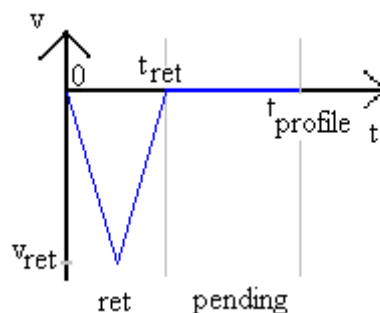
4.5.2 Aikaoptimoitu paluuvaihe

Kun leikkaus suoritetaan materiaalissa olevasta merkistä, vaadittu leikkauspituus saattaa vaihdella muutamasta sentistä jopa kymmeneen metriin. Seuraava leikkausmerkkiä voi olla vaikea ennakoida. Silloin kuljetin on leikkauksen jälkeen palautettava alkuasemaansa mahdollisimman nopeasti, jotta seuraava leikkaus ehditään suorittamaan ajoissa. Palautusta ei kannata kuitenkaan tehdä niin nopeasti, että kuljetin joutuu odottamaan alkuasemassaan jokaista seuraavaa leikkausmerkkiä. Palautus täytyy siis tehdä optimoiden paluujakson ajankäyttöä.

Kuvassa 27 havainnollistetaan aikaoptimoitua paluunopeuden käyttäytymistä. Siinä kuljetin palaa alkuasemaansa ajanhetkellä t_{ret} , mutta seuraava leikkausmerkki ei tulekaan heti, joten kuljetin jää odottamaan alkuasemaansa eli suorittaa taukojakson. Tauko ja käynnissä oleva leikkausprofiili päättyvät ajanhetkenä $t_{profile}$, jolloin leikkausjärjestelmä lukee uuden merkin linjamateriaalista ja uusi leikkausprofiili otetaan käyttöön.

Aikaoptimoidun leikkausprofiilin nopeudenkäyrä eroaa energiaoptimoinnista paluujakson osalta. Aikaoptimoidun paluujakson nopeudenkäyrä suoritetaan kolmion muotoisena, kun taas energiaoptimointi suoritetaan puolisuunnikkaan muotoisella paluunopeuden käyrällä. Kolmion tarkoitus on antaa paluunopeuden absoluuttisen arvon nousta suureksi, sillä siten voidaan pienentää kiihdytyksissä tarvittavia momentteja. Paluujakson kesto on lyhyempi, joten nopeus- ja kiihtyvyysrajat tulevat aikaoptimoinnissa todennäköisemmin vastaan.

Aikaoptimoinnissa sopivan paluuprofiilin määrittävät koneenrakentajat. Heillä on tietämys, minkälaisissa järjestelmissä ja minkälaisilla vaatimuksilla leikkaussovellusta käytetään. Siksi he määrittävät pienimmät leikkauspituudet, joita leikkausjärjestelmässä on mahdollista käsitellä. Pienin käsiteltävä leikkauspituus asetetaan profiilin laskennassa tavoitepituudeksi s_{length} .



Kuva 27. Paluujakson nopeuden profiili ja tauko aikaoptimoinnissa.

Kuljettimen kiihdytys vertonopeuteen

Aikaoptimointia suoritettaessa on mahdollista laskea energiaoptimoinnissa käytetyllä päättelyllä myös aikaoptimoinnin paikan oloarvon määrittäminen sijoittamalla nopeussuhteen arvoksi $\xi = 1$. Nopeussuhteen $\xi = 1$ arvoilla energiaoptimoinnin paluujakson nopeusprofiili vastaa kuvan 27 kolmiota.

Kuljetin liikkuu huippunopeuteen kiihdytyksessä kaavan (4.44) mukaan matkan

$$s_{C,ret,1} = \int_0^{t_{ret,1}} v(t) dt = \int_0^{t_{ret,1}} -a_{ret,1} t dt = -\frac{1}{2} a_{ret,1} t_{ret,1}^2. \quad (4.68)$$

ja linja matkan

$$s_{L,ret,1} = \int_0^{t_{ret,1}} v_L dt = v_L t_{ret,1}, \quad (4.69)$$

Kaavasta (4.49) saadaan paikan oloarvo aikaoptimoinnin huippunopeuteen kiihdyttäessä, kun nopeussuhteen arvo $\xi = 1$:

$$s_C = -\frac{1}{2} s_{C,ret} \left(\frac{s_L}{s_{L,ret,1}} \right)^2. \quad (4.70)$$

Kaavassa s_L on linjan paikan oloarvo kiihdytyksessä ja $s_{L,ret,1}$ linjan kulkema matka paluujakson huippunopeuteen kiihdyttämisessä.

Kuljettimen pysäytys

Vastaavasti kuljettimen pysäytys huippunopeudesta nolla-asemaan onnistuu energiaoptimoinnin kaavalla (4.58), kun nopeussuhteeksi asetetaan $\xi = 1$. Tällöin linjan paikan oloarvon ja kuljettimen paikan oloarvon välille saadaan riippuvuus

$$s_C = -s_{C,ret} \left(\frac{s_L}{s_{L,ret,1}} \right) \left[\frac{1}{2} \left(\frac{s_L}{s_{L,ret,1}} \right) - 1 \right], \quad (4.71)$$

jossa s_L on linjan paikan oloarvo hidastuksessa ja $s_{L,ret,1}$ linjan kulkema matka hidastuksessa. Sen arvo saadaan myös kaavasta (4.69).

Kuljetin palautuu aikaoptimoinnissa alkuasemaansa ennen leikkausprofiilin päättymistä, joten leikkausprofiilin viimeisessä jaksossa odotetaan uutta leikkauskomentoa. Jakson nimi on taukojakso. Käyttäjä ei asetta aikaoptimoinnissa taukojaksolle kestoa vaan se määräytyy odottamisen keston mukaan.

4.5.3 Paluujakson toteutuksen optimointi

Energiaoptimoinnin puolisuunnikkaan ja aikaoptimoinnin kolmion muotoiset paluuprofiilit ovat paloittain määritettyjä. Kaavojen laskeminen monimutkaistaa laskentamallia. Toteutusta voidaan hieman yksinkertaistaa, kun käytetään paraabelin muotoista paluuprofiilia. Paluuprofiili ei ole paloittain määrittely ja laskentaan tarvittavien paikoituspisteiden määrittämisessä voidaan käyttää yhtä nopeuden kaavaa koko paluujaksossa. Kolmas paluuprofiilin optimointitapa on siis toteutuksen optimointi.

Kuljettimen paluuprofiili voidaan tässä tapauksessa määrittää paraabelin toisen asteen yhtälön laskentamallista (Adams 2003). Paraabeli on ylöspäin aukeava ja leikkaa aika-akselin ajanhetkillä 0 ja t_{ret} . Nopeuden huippuarvo v_{ret} löytyy paluuaajan puolivälistä eli hetkellä $\frac{1}{2} t_{ret}$ kuvan 28 tapaan. Koska paraabelin aukeamissuunta on ylöspäin, ja paluunopeus $v_{ret} < 0$, nopeuden toisen asteen yhtälöksi muodostuu

$$v(t) = -4 \frac{v_{ret}}{t_{ret}^2} t(t - t_{ret}). \quad (4.72)$$

Integroimalla nopeus ajan t suhteen saadaan paikan oloarvoksi

$$s_C = \int_0^t v(t) dt = \int_0^t -4 \frac{v_{ret}}{t_{ret}^2} t(t - t_{ret}) dt = -4 \frac{v_{ret}}{t_{ret}^2} \left(\frac{1}{3} t^3 - \frac{1}{2} t_{ret} t^2 \right). \quad (4.73)$$

Linjan paikan ja ajan välillä käytetään riippuvuutta

$$t = \frac{s_L}{s_{L,ret}} t_{ret}, \quad (4.74)$$

missä $s_{L,ret}$ linjan kulkema matka koko paluujaksossa. Huomioimalla tämä lausekkeeseen (4.73) saadaan kuljettimen paikan oloarvoksi

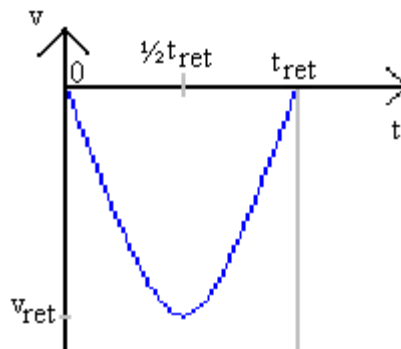
$$\begin{aligned} s_C &= -4 \frac{v_{ret}}{t_{ret}^2} \left[\frac{1}{3} \left(\frac{s_L}{s_{L,ret}} t_{ret} \right)^3 - \frac{1}{2} t_{ret} \left(\frac{s_L}{s_{L,ret}} t_{ret} \right)^2 \right] \\ &= v_{ret} t_{ret} \left(\frac{s_L}{s_{L,ret}} \right)^2 \left[2 - \frac{4}{3} \left(\frac{s_L}{s_{L,ret}} \right) \right]. \end{aligned} \quad (4.75)$$

Paluujakson aikana kuljettu matka tiedetään ja se saadaan myös integroimalla nopeus ajan suhteen. Nopeuden integraalista tulee

$$s_{C,ret} = \int_0^{t_{ret}} -4 \frac{v_{ret}}{t_{ret}^2} t^2 + 4 \frac{v_{ret}}{t_{ret}} t dt = \frac{2}{3} v_{ret} t_{ret}. \quad (4.76)$$

Paluujakson aikana kuljetaan matka $s_{C,ret}$, joten paluujakson suurin tarvittava nopeus on

$$v_{ret} = -\frac{3}{2} \frac{s_{C,ret}}{t_{ret}}. \quad (4.77)$$



Kuva 28. Paluujakson paraabelin muotoinen nopeuden profiili toteutuksen optimoinnissa.

Sijoittamalla tämä viimeisimpään kuljettimen paikan oloarvon lausekkeeseen saadaan lopulta kuljettimelle riippuvuus linjan paikasta:

$$s_C = s_{C,ret} \left(\frac{s_L}{s_{L,ret}} \right)^2 \left[2 \left(\frac{s_L}{s_{L,ret}} \right) - 3 \right]. \quad (4.78)$$

Toteutuksen optimoinnissa paluujakson absoluuttinen maksiminopeus kaavasta (4.38) on suurempi kuin energiaoptimoinnin vastaava paluunopeus

$$v_{ret,0} = \xi v_{ret,1} = -\xi \frac{s_{C,ret}}{t_{ret}} \quad (4.79)$$

kaikilla nopeussuhteen ξ arvoilla. Toteutuksen optimoinnissa suuremman absoluuttisen nopeuden tuottamiseen tarvitaan siis enemmän energiaa, joten energian kulutuksen kannalta tämä ei ole toivottava tapa suorittaa paluujaksoa.

5 Lentävä leikkaus -sovelluksen toteutus ja erikoisvaatimukset

Edellisessä luvussa keskityttiin leikkausprofiilin teoreettiseen puoleen. Tässä luvussa on pyritty tuomaan esiin leikkauksessa käytännössä huomioitavia asioita ja ongelmia. Leikkausjäljen on oltava siisti, eikä leikkauspituuksissa saisi olla vinoumia ja hajontaa johtuen epätarkasta leikkauksesta. On siksi selvitettävä, minkälaisia leikkauspituuksia sovelluksella leikataan tai sahataan, miten suuriksi leikkausajat loppujen lopuksi venyvät ja mitkä seikat vaikuttavat laatuun. Ideoita on saatu haastattelemalla leikkaussovellusasiantuntijoita ja perehtymällä leikkausmateriaaleihin sekä lukemalla sovelluksen toteuttaneiden laitevalmistajien manuaaleja. Kaiken alustavan tietouden pohjalta on myös mallinnettu leikkaussäädön lohkototeutus sovellusohjelmassa.

5.1 Leikkausparametrit

5.1.1 Leikkauspituus

Leikkauspituus voi teoriassa vaihdella lähes nollasta äärettömään. Käytännössä laitteisto, tarve ja tila ratkaisevat, kuinka pitkiä paloja järjestelmällä voidaan tuottaa. Taulukoon 3 on koottu muutamien laitevalmistajien järjestelmiin asetettavia leikkauspituustietoja.

Taulukko 3. Eri laitevalmistajien leikkauspituuksia, jotka on mahdollista suorittaa heidän järjestelmissään.

	Minimileikkauspituus	Maksimileikkauspituus
Katkaisusahaus, Plantool Oy (Salminen 2009)	8 mm	Ääretön
Katkaisusahaus, Iomus Oy (Suontausta 2009)	Ei ole	Ääretön
Pyörösahaus, Contor (Contor 2009)	0,5 m	32 m
Giljotiinileikkuri (Evcon 2009)	24 m	–

5.1.2 Leikkausaika

Leikkausjakson kesto määräytyy materiaalin sahaukseen tai leikkaukseen menevästä ajasta. Leikkausaika vaihtelee riippuen materiaalista, sahausmenetelmästä ja sahaukseen liittyvistä ohjelmista. Jotta leikkaukseen käytetään sopivan pituinen aika, parametriluettelosta määritellään sen kesto joko sahausohjelmista kenttäväylän kautta tai asettamalla manuaalisesti sopiva arvio parametriluetteloon.

Aina leikkausaika ei ole tiedossa tai sitä ei pystytä tarkasti määrittämään. Silloin riittää, että asetetaan käyttöön tarpeeksi pitkä aika leikkauksen suorittamiseen. Taulukkoon 4 on pyritty kokoamaan eri leikkaus-, sahaus- tai katkaisuaikoja. Leikkausajoissa on suuria eroja riippuen leikkausmenetelmästä.

Taulukko 4. Eri laitevalmistajien leikkauksen suorittamiseen liittyviä aikatietoja.

	Katkaisumateriaali	Leikkausaikoja
Katkaisusahaus (Salminen 2009)	Musta materiaaliputki	3–15 s katkaisuaika
	Ruostumaton materiaali	5–40 s katkaisuaika
Giljotiinileikkuri (Evcon 2009)	Teräs	>2,2 s leikkausaika
Katkaisusahaus (Suontausta 2009)	Puu	4,5–5 leikkausta/s (50 mm pitkä lauta) 150–160 leikkausta/min (400 mm pitkä lauta)
Pyörösahaus (Contor 2009)	Ruostumaton teräsputki	5–20 leikkausta/min riippuen leikkaussovelluksesta

5.1.3 Yksiköt ja tarkkuus

ACSM1-laitteessa hyödynnetään sellaisia etäisyyden yksiköitä kuin metri ja tuuma. Leikkauspituus ja muut operatiiviset mitat annetaan kuitenkin hyvin usein millimetreinä. Käyttäjän kannalta on hyvä, jos operatiiviset pituudet voidaan asettaa sovellusohjelmaan millimetreissä, vaikka niitä käsiteltäisiinkin ohjelmassa jossain muussa muodossa.

Nopeussuureista ABB:n laite tukee yksiköitä u/s, u/min ja u/h, jossa u tarkoittaa etäisyydenyksikköä (ABB 2008b). Lentävän leikkurin tapauksessa se tarkoittaa metriä tai tuumaa. SEW:n sovelluksessa linjanopeus voidaan määrittää yksikkönä mm/s (SEW 2005) ja Plantool Oy:n laitteissa yksikkönä m/min (Salminen 2009), jota käytetään myös muualla yleisemmin. Samojen yksiköiden käyttö kaikkialla ohjelmassa on yhtäläisyyden kannalta hyvä tapa, mutta tässä sovelluksessa on kenties parempi tehdä poikkeus ja asettaa leikkauspituus sekä muut operatiiviset pituusmitat millimetreissä ja nopeussuureet yksikkönä m/s tai vielä mieluummin m/min.

Taulukoon 5 on koottu muutamien laitevalmistajien järjestelmissä käytettäviä tarkkuuksia ja nopeustietoja. Kilpailijoiden sovelluksissa puolestaan mainitaan hyvin usein, että leikkauspituus voidaan määrittää 0,001 yksikön tarkkuudella. Se, mitä yksiköillä tässä tarkoitetaan, jää kuitenkin usein epäselväksi. Toisaalta, jos yksiköt on ilmoitettu, niin sitten tarkkuus on jätetty mainitsematta. Tutkimalla taulukoiden 3,5 ja 6 tietoja huomataan tarkkuuden olevan hyvin usein 0,1 käytettävän yksikön ollessa millimetri. Tätä tarkkuutta voidaan pitää myös vaadittavana ABB:n sovellukseen.

Taulukko 5. Eri laitevalmistajien lentävä leikkaus -sovelluksessa käytettäviä linjanopeuksia ja leikkaustarkkuuksia.

	Linjanopeus	Leikkaustarkkuus
Plantool Oy	0–10 m/min. On tulossa järjestelmiä, joissa nopeus voi olla 50 m/min.	-
Iomus	-	0,2 % leikattavan kappaleen pituudesta.
Evcon	9,5 m/s	+/- 50 mm
Contor	Mallista riippuen 30–120 m/min välillä.	+/- 3 mm
Gefran	60 m/min	0,3–0,5 mm

5.1.4 Materiaalihukka ja sahausrako

Sahaavissa leikkauksissa materiaalia menee leikkauksessa hukkaan, sillä materiaalia lastuuntuu leikkausterän leveyden verran. Riippuen leikkaustarkkuudesta, jolla leikkauspituudet määritetään, on huolehdittava, että leikkausraon leveys ei lyhennä leikkausmateriaalin tavoitepituutta. Siksi sahausraon ja leikkausraon leveys on huomioitava leikkausta tehtäessä. Sahauksessa, katkaisussa ja leikkauksessa muodostuvan raon leveys vaihtelee tapauskohtaisesti. Taulukkoon 6 on koottu esimerkkimielessä joitakin arvoja.

Sahausraon leveyden voi sisällyttää suoraan leikkauspituuteen, mutta tällöin on todennäköistä, että sovelluksen käyttäjät eivät muista ottaa huomioon sen vaikutusta tavoitepituuteen. Leikkausraon leveyttä kontrolloiva parametri on hyvä lisätä sovellusohjelmaan. Parametrin tarkoituksena on huolehtia tarkkuudesta ja toimia muistuttimena. Markkinoilla olevien laitevalmistajien sovelluksissa ei ole erikseen parametria leikkausraon leveydelle.

Joissakin sahaustyökaluissa ei riitä, että pelkästään määrittää leikkaus- tai sahausterän leveyden. Esimerkiksi pyörösahoissa ilmenee sahan värähtelyä, joka leventää muodostuvaa sahausrakoa. Värähtelyn vaikutuksesta syntyvä sahausraon levenemä ei ole määritettävissä teoreettisesti, vaan se on määritettävä tekemällä sahaus käsiteltävällä materiaalilla ja mitattava muodostuneen sahausraon leveys siitä. Kyse on vain muutamista millimetreistä, mutta värähtelyn kasvattava vaikutus on kenties joissakin tapauksissa otettava huomioon (Kotilahti 2009).

Taulukko 6. Eri sahausmenetelmillä ja leikkausmateriaaleilla muodostuvat leikkausraot.

Sahausmenetelmä	Sahausmateriaali	Terämateriaali	Terän leveys (mm)
Katkaisusahaus (Salminen 2009)	Ruostumaton metalli ja musta rauta	Kovametalli	1,7–3,4
		Pikateräkatkaisu	2,5–8
Katkaisusahaus (Suontausta 2009)	Puu		Yleensä 3,2
Vannesahaus (Varis ja Tanskanen 2006, Kotilahti 2009)	Terästanko	Hiili, pikateräs tai kovametalli	2–4,5
Pyörösahaus (Kotilahti 2009)	Puu		Yleisesti hieman suurempi kuin vannesahauksessa. Suurenee lisäksi värähtelyn vaikutuksesta 0–0,2 mm
Laserleikkaus (Anon 2004)	Kartonki		Noin 0,1

5.2 Muita sovellusteknisiä näkökohtia

Takaisinkytkennät

Lentävässä leikkauksessa on tarve ainakin kahdelle takaisinkytkennälle. Toinen takaisinkytkennöistä mittaa linjanopeutta ja toinen kuljettimen omaa paikkaa. Näistä saatujen tietojen pohjalta voidaan rakentaa perusliikeprofiili.

Takaisinkytkentää voidaan tarvita myös leikkausmateriaalin mahdollisen leveyden muuttumisen tarkkailuun leikkauksen aikana, leikkauspalojen laadunvalvontaan ilmoittamaan käytölle vikaantuneet tai hyväksytyt leikkauspalat ja leikkauskomennon aloituksen määrittämiseen.

Profiilin vaihto lennosta

SEW:n ohjausohjelmaan on mahdollisuus asettaa kahdeksan eri leikkauspituutta, jotka vaihdetaan ohjelmassa binäärikoodia muuttamalla. Leikkauspituuden vaihto ei kuitenkaan onnistu leikkauksen toimiessa (SEW 2005). Liitteessä A olevia muiden kilpailijoiden lentävä leikkaus -toimintoja tutkimalla huomataan, että kovin monessa kilpailijoiden laitteissa ei tueta profiilin vaihtoa. Profiilin vaihdon toteuttaminen olisi näin ollen kilpailuetu.

Parametrisoitavaan ohjelman rakenteeseen on mahdollista liittää useita leikkauspituuksia, jos ne tallennetaan taulukkoon. Taulukoiden leikkauspituuksia voidaan hyödyntää osoittamalla haluttuun taulukkoalkioon. Taulukkoalkioihin osoittaminen voidaan vaihtaa sovellusohjelmasta manuaalisesti tai kääntämällä digitaalisia sisääntulokytkimiä. Vaihdos voidaan tehdä leikkauksen ollessa käynnissä, mutta uusi leikkauspituus astuu voimaan vasta seuraavan leikkauksen käynnistyessä.

Nokkapyörätyökalulla on mahdollista ladata 32 nokkapyöräprofiilia käyttöön. Profiilien keskinäinen vaihtaminen on mahdollista, jos profiilien vaihto on esiasennettu kuhunkin

profiiliin. Niitä voidaan pyörittää jatkuvasti tiettyinä sekvensseinä, kuten 1 2 3 ja 1 2 3, 1 1 1 ja 1 1 1 tai 1 3 ja 1 3, mutta ei kuitenkaan järjestyksessä 1 3 2. Uusien profiilien lataaminen käyttöön onnistuu vain pysäyttämällä leikkausprosessin. Nokkapyörotoiminnallisuus ei kuitenkaan ole suositeltava menetelmä leikkauksen toteuttamiseen.

On kuitenkin hyvin todennäköistä, että asiakkaat eivät tarvitse monimutkaisia sekvenssejä. Tarve on ennemminkin jonkinlaiselle suorituslaskurille, jolla määritetään, montako kappaletta tietyn mittaisia paloja leikataan. Leikkauspituus pysyy vakiona, kunnes haluttu kappalemäärä jotakin pituutta saadaan valmiiksi ja siirrytään suorittamaan seuraavaa leikkauspituutta tietyllä kappalemäärällä.

Laskurit

Laskurit ovat tarpeellisia jatkuvan materiaalin käsittelyssä, jotta tiedetään kuluneen materiaalin määrä ja voidaan laskea leikattujen kappaleiden määrä. Jos sovellusta käytetään lentävän leikkauksen sijasta muuhun tarkoitukseen, kuten pullon täyttöön, voidaan laskurilla ilmoittaa täytettyjen pullojen kappalemäärä. Taulukoon 7 on listattu kahdeksan eri laskuritoimintoa, jotka voivat olla tarpeellisia sovellukseen. Laskurin toteutuksessa laskennan aloitus aloitetaan joko ensimmäisestä leikkauksesta tai vasta toisesta leikkauksesta, jos ensimmäisen leikkauksen tarkoitus on poistaa huonoa materiaalia.

Taulukko 7. Sovelluksessa mahdollisesti tarvittavia laskureita (Lenze 2008).

Laskurityyppi	Selitys toiminnalle
Laskuri pituuden mukaan	Laskuriin lisätään aina täsmälleen yksi leikkaus, kun tavoitemittaisen kappaleen leikkaus on suoritettu. Toiminto edellyttää, että säädin saa signaalin ”leikkaus suoritettu”.
Laskuri merkistä	Laskuriin lisätään aina täsmälleen yksi leikkaus, kun käyttö saa leikkauskomennon eli käynnistyssignaalin.
Kappalelaskuri	Valmiiden kappaleiden laskeminen voidaan suorittaa ulkoisella liipaisulla jonkin anturin avulla. Kappalelaskuriin lisätään yksi, kun anturi havaitsee jonkin tietyn visuaalisen merkin. Toiminnallisuus on hyödyllinen järjestelmäkäytöissä, joissa suoritetaan laadunvalvontaa.
Hylkäyspalojen laskuri	Jos leikkauspala hylätään syystä tai toisesta leikkauksen jälkeisessä laadunvalvonnassa, käytölle lähetetään tieto tästä. Samalla hylkäyspalojen laskuriin lisätään yksi hylkäys. Toiminto edellyttää laadunvalvontaa.
Vähennyslaskuri	Jos leikkauspala ei jostain syystä voida hyödyntää, laskurista voidaan vähentää yksi leikkaus tämän toiminnon avulla. Vähennyslaskuri vähentää automaattisesti yhden, jos pituusleikkuri vikaantuu jostain syystä.
Suoritelaskuri	Suoritelaskuriin asetetaan haluttu määrä leikkaussuoritteita. Kun ensimmäinen leikkaus on suoritettu, vähennetään suoritelaskurista yksi ja jälleen seuraavan leikkauksen jälkeen yksi, kunnes suoritelaskurin arvoksi tulee nolla. Jos tapahtuu virhe tai jokin leikkauspala hylätään, suoritelaskuriin lisätään yksi.
Nollaus	Jokainen laskuri voidaan nollata halutussa tilanteessa tai jos haluttu määrä kappaleita on saatu leikatuksi. Laskuri nollataan, jos leikkaustoimitilaa vaihdetaan prosessoinnin aikana. Poikkeuksena on kuitenkin vähennyslaskuri.
Laskurin käyttöönotto	Laskuria ei kaikissa tapauksissa tarvita, joten laskuri voidaan määrittää päälle tai pois päältä tilaan.

Materiaalimuutosten vaikutus leikkausaikaan

Jos linjamateriaalin leveys, paksuus tai halkaisija muuttuu, on tarpeellista asettaa jokin anturi tarkkailemaan muutosta. Muutosten tarkkaileminen ei ole kuitenkaan suoranaisesti kuljettimen liikettä tarkkailevan käytön tehtävä, vaikka materiaali muutokset on otettava sielläkin huomioon. Leveyden, halkaisijan ja paksuuden muutokset vaikuttavat leikkausaikoihin joko lyhentävästi tai pidentävästi. Leikkausjärjestelmässä muutosten tarkkailu on tärkeämpää, jotta osataan antaa leikkausjärjestelmälle oikeat työstövoimat.

Leikkausaika voidaan saada suoraan kenttäväylän kautta leikkausta ohjaavalta järjestelmältä tai määrittää suoraan kuljetinta ohjaavalta käytöltä manuaalisesti. Manuaalisesti asetettaessa aikaa kannattaa varata paksuimman ja leveimmän materiaalikohdan leikkausajan mukaan. Materiaalimuutosten ja sitä kautta leikkausajan määrittäminen suoraan leikkausjärjestelmästä on selvästi harvinaisempi tapa, mutta

ainakin joissakin automaattisissa vannesahoissa on automaattinen materiaalin leveyden mukaan säätyvä teräohjain (Varis ja Tanskanen 2006).

Rinnakkainen leikkaus

Järjestelmäkäytössä voi olla sahausteriä myös kaksi tai vieläkin useampia, jolloin sahausterät toimivat rinnakkain. Kahden sahanterän tapauksessa toinen leikkausterä leikkaa, kun toinen leikkausterä esimerkiksi hakeutuu laitteen ohjauksen määrittelemään optimaaliseen uuteen asemaan. Rinnakkaisella sahauksella voidaan nostaa linjan tuottavuutta.

ACSM1-laitteessa kuljettimien rinnakkaisohjaus ei onnistu yhdellä taajuusmuuttajakäytöllä, sillä taajuusmuuttaja ei voi ohjata kahta moottoria kerrallaan. Lisäksi takaisinkytkentäpaikkoja ei ole tarpeeksi jokaisen kuljettimen paikan seurantaan eikä suorituskapasiteetti riitä usean kuljettimen ohjaamiseen. Rinnakkainen leikkaaminen voi onnistua siten, että jokaista kuljetinta kohden on oma taajuusmuuttajakäyttö ja ne kommunikoivat keskenään.

Jokaisella liikettä ohjaavalla käytöllä on omat paikan takaisinkytkennät, mutta linjanopeustiedot ja leikkauspituudet voidaan käsitellä yhdessä käytössä ja sieltä edelleen ohjata muiden käyttöjen liikkeenohjaussäätimille. Jonkinlaiset vuorotteluohjeet ja orjakäytön käynnistys viivästysohjeet tarvitsee kenties ohjelmoida, mutta rinnakkainen leikkaus on hyvä toteuttaa vasta, kun leikkaus saadaan toteutettua onnistuneesti yhdessä käytössä ja nähdään järjestelmäkäytön tarpeet paremmin. Rinnakkaisen leikkauksen tutkiminen ei kuulu tämän työn sisältöön.

Usean kappaleen leikkaus samanaikaisesti

Leikkausjärjestelmän suoritustehokkuutta voidaan parantaa myös leikkaamalla leikkausmateriaalista samanaikaisesti useita kappaleita. Usean kappaleen leikkaaminen lentävässä leikkauksessa on mahdollista, mutta silloin on kiinnitettävä erityistä huomiota materiaalien liikkuvuuteen. Tällöin on puristettava paininrullilla leikkausmateriaalia kovempaa liikkuvuuden minimoimiseksi, jolloin taas materiaalin pinta voi helpommin vaurioitua.

Sahaussovelluksissa ja muita kovia materiaaleja leikattaessa tapahtuu aina pieni nykäisy, kun leikkaustyökalu läpäisee leikkausmateriaalin. Tämä voi häiritä vielä sahaamattomien sahauspalojen sahausta ja vaurioittaa niitä sahauspaloja, joita ei vielä ole saatu sahattua poikki. Seurauksena on sahausjäljen laadun huonontuminen materiaalissa.

Etäkäyttömahdollisuus

Järjestelmäkäytön etäkäyttö, parametrien asettelu ja järjestelmän tarkkailu on mahdollista mistä tahansa, jos järjestelmäkäyttö on kytketty verkkoon. Tämä on mahdollista käyttämällä tiedonsiirrossa Ethernet-väyläprotokollaa. ACSM1-laitteeseen on mahdollista lisätä Ethernet-liitäntä optiona Muista laitevalmistajista (Lenze 2008) ja Yaskawa (2006) tukevat ethernet-liitäntää laitteissaan. Kukaan kilpailijoista ei kuitenkaan käytä Ethernet-protokollaa lentävä leikkaus -järjestelmissä, ja tällä hetkellä sillä on vähän arvoa sovelluksen toteuttamisessa.

Sovelluksen ja Ethernet-protokollan kehittyessä Ethernetin käyttö tulee olemaan todennäköistä, koska sillä on mahdollista korvata sarjamuotoisia kenttäväyläprotokollia ja I/O-liitäntöjä. Lisäksi suuntauksena on, että kenttäväylät ovat hiljalleen korvautumassa Ethernetillä (Himanshu 2006). Ethernet käyttää standardoitua tekniikkaa, on luotettava, nopea ja skaalautuva kaikkein vaativimpiinkin

liikkeenohjaussovelluksiin. Sen avulla on myös mahdollista käyttää yhtä ja samaa verkkotekniikkaa läpi koko järjestelmän (Silvola 2006).

5.3 Sovelluksen vikatilanteet

Laitteen käyttöönotossa ja käytössä voi ilmetä vikatilanteita. Kaikkiin mahdollisiin ohjelman suorituksessa ilmenevin vikatilanteisiin ei puututa, mutta tietyn mittaisen leikkauspalkan leikkaamiseen liittyy kuitenkin näkökohta, joka on hyvä nostaa esiin. Saattaa nimittäin olla, että järjestelmä ei havaitse jostain syystä leikkausmerkkiä tai pysty saavuttamaan alkuasemaa ennen seuraavan leikkauksen alkua. Taulukossa 8 on kolme eri vaihtoehtoa, miten tilanne voidaan kenties hoitaa.

Kaksinkertaisen pituuden tuottaminen on siinä mielessä hyvä ratkaisu, sillä leikkauspala voidaan katkaista jälkikäteen, eikä siten muodostuisi ylimääräistä materiaalihukkaa toisin kuin hätäleikkauksessa ja linjanpysäytyksessä. Kuitenkin esimerkiksi Plantool Oy:ssä käytetään linjapysäytystä vikatilanteen sattuessa. Syynä tähän saattaa olla se, että yleisin vikatilanne on terän rikkoutuminen, jolloin järjestelmä on pysäytettävä (Roos 2009). Keskeytymisen syytä voi yrittää myös korjata pienentämällä linjanopeutta, suurentamalla kiihtyvyyden raja-arvoja tai suurentamalla matkaa, jonka kuljetin voi liikkua.

Leikkaussovelluksen käyttöönoton yhteydessä kaikki mahdolliset ongelmat pyritään välttämään sovellusohjelman alustustoimilla, jolloin tarkoitus on tehdä tarkistuslaskuja ja asettaa parametrirajat. Näiden avulla tarkistetaan, että leikkausprofiili järkevä suorittaa. Jos ei, niin sovellusohjelma antaa tästä ilmoituksen, eikä suostu edes käynnistymään.

Taulukko 8. Tavoitemittaisen leikkauspalkan leikkauksen keskeytymisen jälkeen suoritettava toiminto.

Vaihtoehto	Leikkauksen keskeytymisen jälkeinen toimintoehdotus
1	Järjestelmä jää odottamaan alkuasemaan seuraavaa leikkausta, jolloin työstettävän kappaleen pituudesta tulee kaksinkertainen.
2	Tehdään hätäleikkaus, palautetaan kuljetin alkuasemaan ja lähdetään suorittamaan seuraavaa leikkauskomentoa. Pohjimmainen ongelma voi kuitenkin olla se, että leikkauspituus on niin lyhyt, ettei leikkausta kerta kaikkiaan ehditä suorittamaan. Tällöin useita leikkauksia peräkkäin hylätään.
3	Pysäytetään koko linja, mutta silloinkin on vaarana, että järjestelmän alkuosassa tapahtuu vaurioita.

5.4 Työsuojelua koskevia näkökohtia

Sahaus- ja leikkaustyökaluja käytettäessä on vaara, että syntyy materiaali- tai henkilövahinkoja. Järjestelmään on siksi asennettava hätäpysäytysluokan edellyttämät hätäpysäyttimet ja muut tarvittavat lisälaitteet, jotka kytetään taajuusmuuttajakäyttöön. Järjestelmän tulee pysähtyä nopeusramppien mukaan, kun hätäpysäytyssignaali on havaittu (ABB 2008b). Tämän jälkeen katkaistaan virran syöttö. Jos hätäpysäytyssignaali vastaanotetaan leikkaamisen aikana, leikkaustyökalun kuljettimen on seurattava linjan nopeutta pysähtymiseen asti (Hedman 2009). Leikkausterä on osittain materiaalin sisällä, joten leikkausvaiheen voimakkaat nopeuserot aiheuttavat tuhoja järjestelmäkokonaisuudessa.

Hätäpysäytyssignaalin käsittelyn yhteyteen voi olla tarpeellista liittää joitakin toimintoja, joiden tarkoitus on muun muassa säilyttää tiedot edellisestä istunnosta. Lisäksi järjestelmäkäytön voidaan haluta tekevän jotakin hätäsignaalin poiskytkemisen yhteydessä. Mitään yleiskäyttöisiä toimintoja ei voida määrittää, sillä toiminnot määräytyvät tapauskohtaisten järjestelmien mukaan.

ACSM1-laiteessa on SIL3-tasoinen Safe-Torque-Off -turvafunktio (ABB 2007), joka estää tehovuolijohdekomponentteja saamasta ohjausjännitettä. Toiminto estää vaihtosuuntaajaa luomasta moottorin käynnistämiseen tarvittavaa jännitettä. Toiminnon ansiosta koneen muiden kuin sähköosien lyhytaikaiset puhdistus- ja huoltotyöt on mahdollista suorittaa katkaisematta virran syöttöä taajuusmuuttajaan (ABB 2008b).

5.5 Lentävä leikkaus -säädön periaate

Lentävä leikkaus -säätö toteutetaan lisäämällä ACSM1 taajuusmuuttajakäytön kiinteään ohjelmistoon sovellusparametrit ja sovellusohjelmaan teknologialohkot, jotka sisältävät kaikki sovelluksen tarvitsemat toiminnot. Lisättävien teknologialohkojen toiminnot esitellään vain korkealla tasolla, eikä puututa lohkojen sisäiseen toteutukseen. Liitteessä C on lentävä leikkaus -sovelluksen arkkitehtuuri lohko-ohjelmasta, ja lentävä leikkaus -lohkojen kytkeytyminen osaksi laitteen muuta ohjausjärjestelmää. Liitteessä on myös esitetty tarkemmin eri osa-alueisiin liittyvät piirteet.

Lentävä leikkaus -säädössä liikettä ohjataan jaksoihin jaetulla liikeprofiililla. Jokaiselle jaksolle määrätään oma paikan oloarvon laskentakaava, jota käydään läpi useita kertoja jakson aikana paikoituspisteiden määrittämistä varten. Säädin interpoloi pisteistä leikkausprofiilin, kunnes jakson tavoite on saavutettu. Tämän jälkeen siirrytään seuraavaan jaksoon, kunnes kaikki kuusi jaksoa on käyty läpi. Leikkaus voidaan aloittaa tämän jälkeen jälleen alusta.

Paikoitukseen liittyvät vakiolohkot käyttävät 500 μ s:n aikatasoa (ABB 2008b), eli lohkojen rajapinnan taakse kätkeyt algoritmit suoritetaan 500 μ s:n välein. Nopeammalla aikatasolla lohkoja ei kannata suorittaa, vaikka se onkin mahdollista tehdä 250 μ s:n aikatasolla, sillä saatua paikkainformaatiotietoa ei pystytty hyödyntämään. Seurauksena on suorituskyvyn tuhlaantumista. Tällainen tilanne on esimerkiksi teknologiakirjastosta lisättävällä nokkapyörätoiminnallisuuden cam-lohkokolla, joka voidaan suorittaa 250 μ s:n aikatasolla.

Lentävän leikkauksen kannalta aikatazon valinta vaikuttaa lähinnä vain leikkaustarkkuuteen. Mitä hitaampaa aikatasoa käytetään, sitä enemmän tingitään leikkaustarkkuudesta. Leikkaussäädössä profiilinluontilohkoa eli lohkoa, joka käy eri jaksojen paikoituskaavoja läpi, halutaan suorittaa mahdollisimman nopealla aikatasolla. Sopivan aikatazon määrää kuitenkin suorituskyyä eniten kuormittava jakso eli jakso, johon liittyy eniten ehtoja ja kaavojen monimutkaisuutta.

Aikataso valitaan siten, että suorituskyyyn kannalta kriittisin profiilinluontijakso saadaan vielä hyvin suoritettua. Aluksi valitaan aikatasoksi 2 ms, joka on myös Lenzen käyttämä aikataso (Lenze 2008). Aikatasoa voidaan vielä nopeuttaa, jos sille tulee tarve. Takaisinkytkennän paikkatieto ja käynnistysinformaatio päivittyvät sovellusohjelmassa myös 500 μ s:n ja 250 μ s:n välein.

Lohkot, joihin syötetään parametritietoja, käydään läpi vain kerran leikkausprofiilin päätteeksi taukojaksossa tai laitteen ensimmäisen kerran moduloidessa. Parametreja voidaan muuttaa leikkauksen aikana, mutta ne astuvat voimaan vasta seuraavassa leikkauksessa. Jos useita leikkausparametreja halutaan muuttaa kerralla, niitä ei välttämättä ehditä tekemään yhden leikkauksen aikana. Tällä voi olla vaikutusta leikkauksen tavoitteeseen, ja pahimmillaan leikkaus voi keskeytyä virheellisten parametritietojen johdosta.

6 Pohdinnat ja johtopäätökset

Työn yhtenä tavoitteena oli tutkia kilpailijoiden lentävä leikkaus -sovellusta. Lentävä leikkaus -sovellus löytyy monilta taajuusmuuttajavalmistajilta, kuten liitteen A taulukoista käy ilmi. Lisäksi markkinalaajuusanalyysitutkimus osoitti, että leikkausjärjestelmillä on muun muassa jo toteutettua kelainsovellusta suurempi markkinaosuus. Siksi leikkaussovelluksia, niin lentävää leikkuria kuin pyörivää leikkuriakin, voidaan pitää seuraavina potentiaalisina sovelluksina ACSM1-laitteelle.

Työn päätavoitteena oli lentävän leikkauksen toteutus taajuusmuuttajasovelluksena. Toteutusta on lähdetty kehittämään nokkapyörätoiminnallisuudella, joka on osittain valmis. Nokkapyörätoiminnallisuuden jäykkyys, tavoitemittaisten leikkauspalojen saavuttamiseen liittyvä ongelma ja muiden laitevalmistajien toteutuksien tutkiminen kuitenkin osoitti, ettei nokkapyörätoiminnallisuus sellaisenaan ole paras mahdollinen lähestymistapa sovelluksen toteuttamiseksi. Muiden laitevalmistajien suosimasta menetelmästä parametrisointi osoittautui toimivimmaksi ratkaisuksi toteuttaa sovellus.

Nokkapyörätoiminnallisuudessa voidaan luoda muutamia mielivaltaisen muotoisia liikeprofiileja ja niiden välille voidaan luoda joitakin yksinkertaisia sekvenssejä. Yhteen leikkaukseen vaadittava liikeprofiili ei kuitenkaan mahdu yhteen taulukkoon, vaan se on jaettava epäloogisesti usealle peräkkäin ajettavalle taulukolle. Leikkauspituus voi olla niin suuri, ettei leikkausprofiili mahdu paloitetunakaan taulukoihin. Väilyssuhdetta käyttämällä leikkausprofiili voidaan saada mahtumaan jopa yhteen taulukkoon, mutta sen käyttö on hyvin monimutkaista ja väilyssuhdetta on vaihdettava jatkuvasti.

Parametrisoinnissa sovellusohjelmaan lisätään parametreja, joiden pohjalta leikkaussäädin luo leikkausprofiilin. Parametrisoitavassa lähestymistavassa leikkausprofiili jaetaan säädön kannalta kiihdytykseen, paikoituksen korjaukseen, leikkaamiseen linjan tahdissa, jarrutukseen, palautukseen ja taukoon. Leikkausprofiilin jaottelu mahdollistaa erilaisten toiminnallisuuksien liittämisen eri kohtiin leikkaussovelluksessa ja tekee toteutuksesta joustavamman. Erityisen kiinnostavaksi jaottelussa osoittautui paluujakso, johon löytyi useita toteutusvaihtoehtoja.

Parametrisoitavaan profiilinluontitapaan liittyy tietysti myös joitakin ongelmia, jotka vaativat vielä hiomista ja joihin ei tässä työssä ole pystytty esittämään tyydyttävää ratkaisua. Erityisen ongelmalliseksi osoittautui linjan paikkatiedon nollaaminen leikkausjakson päättyessä. Se ratkaistiin työssä menetelmällä, jossa paikkatietoa ei nollata vaan siitä vähennetään leikkauspituus leikkausjakson päätteeksi. Menetelmä on toimiva, sillä paikkatieto ei pääse kasvamaan eikä paikkatiedon raja-arvo tule vastaan. Toinen ongelma liittyi liikeprofiilin vaihtamiseen leikkausprosessin ollessa käynnissä. Se oli toivomuksena sovelluksen toiminnallisuuksiin. Leikkausprofiilin vaihtaminen onnistuu kyllä asettamalla parametreista uusi leikkauspituus. Mitä enemmän haluaa kuitenkin muuttaa muita sovellusparametreja, sitä epätodennäköisempää on, että se onnistuu yhden leikkausjakson aikana.

Leikkausprofiili laskee reaaliaikaisesti paikoituspisteitä ja interpoloi pisteistä leikkausprofiilin. Taajuusmuuttajakäytön kannalta pisteiden laskeminen voi kuitenkin kuormittaa suorituskykyä liikaa. Vielä yksi vaihtoehto leikkauksen toteuttamiseen on laittaa leikkausoperaatioon liittyvät tiedot laitevalmistajan CT tapaan dynaamisiin etukäteen laskettuihin jaksoitettuuihin taulukoihin, jolloin niiden laskentaan ei tarvita suorituskykyä niin paljon. Profiilin paluujaksoon käytettäisiin paikoituspisteiden

interpolointia. Työn tavoitteena ei kuitenkaan ollut suorituskyvyn tutkiminen, joten se jätetään tuotekehityksen harkintaan.

Tässä työssä ei tutkittu varsinaisesti teknistä yksityiskohtaa eikä sitä kohdistettu mitään yksittäistä tapausta silmällä pitäen. Siksi työn sisältö keskitettiin yleiselle tasolle ja yleiskäyttöisyyttä silmällä pitäen. Asiakaskontaktien luonti onkin tulevaisuudessa tärkeää sovelluksen kehityksen kannalta. Työssä esiteltyä paikoituskonseptia on jo kokeiltu onnistuneesti eräällä asiakkaalla Italiassa. Asiakaskontakteja on syytä olla kuitenkin enemmän, sillä sovelluksen tarpeet ilmenevät parhaiten kentältä. Tämä työ on tuotekehitysprojektilla esiselvitys, jossa pyrittiin minimoimaan edessä olevia ongelmia, keräämään mahdollisimman paljon tietoutta kilpailijoista ja sovelluksesta sekä hahmoteltiin lentävä leikkaus -sovellus osaksi muuta säätöjärjestelmää.

Lopuksi yhteenvetona luetellaan joitakin vaatimuksia ohjelman toteuttamiseen, jotta sovellus saadaan menestymään markkinoilla. Linjalta ja leikkurin kuljettimelta tarvitaan takaisinkytkennät paikoitusohjeiden laskentaa varten. Paikoitusohjeiden määrittämiseen tarvitaan sovellusohjelmassa paikoituskäskyjä, joita on useita johtuen leikkauksessa tarvittavan profiilin jakamisesta jaksoihin. Perusjaksoja on kuusi, mutta esimerkiksi paluujaksoja on kaksi: energia- ja aikaoptimointi. Käytännössä tarvittavia jaksoja muodostuu siis enemmän.

Leikkauspituuden määrittämisessä on otettava huomioon leikkauspituuden lisäksi leikkausraon vaikutus ja mahdollisuus usean leikkauspituuden käyttämiseen ohjelmasta. Leikkauspituuksia voidaan vaihtaa joko ohjelman taulukoihin tallennettuja leikkauspituuksia muuttamalla tai muuttamalla käytössä olevaa leikkauspituustietoa lennosta. Leikkauspituus tulisi saada asettaa 0,1 mm:n tarkkuudella, linjanopeuden yksikkönä ollessa m/s, m/min tai jopa mm/s.

ACSM1-laite sisältää vakiona valtavat määrät toiminnallisuuksia, hätäpysäytysluokan vaatimat toiminnot ja vikatilanteiden varalle viankäsittelyt. Näitä kaikkia voidaan hyödyntää lentävässä leikkauksessa laajentamalla ne kattamaan sovellusohjelman tarpeet. Lisää ideoita voidaan kerätä myös liitteen A taulukosta. Sellaisia voivat olla esimerkiksi laskurit, nykäystoiminto ja vino leikkauskulma.

7 Lähteet

- ABB (2006): "AC drives help to boost roll plant efficiency in bakeries", revision A (EN), application notes, ABB library, ABB motors&drives.
- ABB (2007): "Application Guide Safe Torque Off Function (STO)", revision C (EN), ACSM1 Drive Manuals, ABB Library, ABB/Drives 3AFE68929814.
- ABB (2008a): ABB Drives – Technical guide book, revision D, Tekninen opas, ABB Library, ABB/Drives 3AFE64514482.
- ABB (2008b): "ACSM1 Motion Control Program Firmware Manual", revision C (EN), Ohjelmointiopas, ABB Library, ABB/Drives.
- Adams R. (2003): "Calculus – A complete course." 5. painos. Kanada: Addison Wesley Longman. ISBN 0-201-79131-5.
- Anon (2004): VTT:ltä kartongeille laserleikkausmenetelmä. Metallitekniikka, 2004, vol. 57:9, s. 44. ISSN 1237-6663.
- Anon (2009): Prolaser Oy, Riihimäki: laserleikkaus, vesileikkaus, särmäys [Viitattu 10.9.2009]. Verkko-osoite: <http://www.prolaser.fi/index.html>
- Artto, K. (2008): "Projektiliiketoiminta", Helsinki, WSOY Oppimateriaalit Oy, 2. painos, ISBN 978-951-0-31482-1.
- Cantor (2009): Shengong Cantor Machine & Electric Equipment Co., Ltd, Kiina. Flying saw|pipe cutting-making-cutoff machine|Spiral accumulator|shear welder|welded pipe machi [Viitattu 29.10.2009]. Verkko-osoite: http://en.scontor.com/index_en.aspx.
- CT (2001): Gareth, J. "Flying Shear Application Software", 1. painos, 59 s., osanumero: SSP-3100-0285.
- Evcon (2009): EVO TECH PVT. LTD. Intia. Evo Tech Pvt.ltd.: Flying shear, Rolling Mill, Blast Frnace Consultancies, Evcon Turbo TMT [Viitattu 29.10.2009]. Verkko-osoite: http://www.evotech.in/inview/index.php?catagory=evcon_flying_shear.
- Hedman A. (2009): Kotimaan myynnin tuotepäälikkö. ABB Oy. Valimopolku 4A, 00380 Helsinki. Sähköpostikeskustelu, 26.10.2009.
- Himanshu S. (2006): "General Motion Control Worldwide Outlook", Market analysis and forecast through 2010, ARC Advisory Group, 180s., [Viitattu 16.12.2009].
- Kotilahti, T. (2009): Puun työstön teoria. Luentomateriaali, Teknillinen korkeakoulu, Puunjalostustekniikan laitos. Espoo.
- Lehtinen L. (2008): Lineaarimoottori yksinkertaistaa vesileikkurin. Tekniikka&Talous – 15. helmikuuta 2008 Teema automaatio, lehden numero 6/08, s. 5. ISSN 0785-997X.
- Leino, R. (2006): Taajuusmuuttaja saa lisää älyä, Tekniikka&Talous, lehden numero 11/2006, s. 22-23. ISSN 0785-997X.

Lenze (2008): "Prepared solution, FlyingSaw", [Verkkomanuaali], versio 1.1, 03/2008, Lenze IEC61131-3 inside [Viitattu 2009]. Saatavissa: http://www.lenze.com/downloadbereich/en/B1_Servo_Technology/B18_Prepared_Solutions/SW_FlyingSaw_v1-1_EN.pdf

Luomi, J. (2008): Sähkökäyttöjen suunnittelu, Luentomoniste, Teknillinen korkeakoulu, Sähkötekniikan laitos, Espoo.

Lönnberg, M. (1995): Selvitys uuden taajuusmuuttajan soveltuvuudesta metallilangan vetovalmistukseen. Diplomityö, Teknillinen korkeakoulu, sähkötekniikan osasto. Helsinki. 62 s.

Mäkeläinen M. (2009): ACSM1 Tuotepäällikkö. ABB Oy. Valimopolku 4A, 00380 Helsinki. Keskustelu, 15.12.2009.

Niiranen, J. (1997): Tehoelektroniikan komponentit, Helsinki, Otatieto Oy, ISBN 951-672-241-5.

Roos P.(2009): Automation manager. Plantool Oy. Verkatehtaankatu 10, 60100 Seinäjoki. Sähköpostikeskustelu, 9.11.2009.

Salminen, K. (2009): Product Group Manager, Special Machine Tools. Plantool Oy. Verkatehtaankatu 10, FI-60100 Seinäjoki. Sähköpostikeskustelu, 2.9.2009.

SEW (2005): "MOVIDRIVE MDX61B Flying Saw Application Manual". [Verkkomanuaali]. painos 08/2005, Vaihdemoottorit\Teollisuusvaihteet\Käyttötekniikan elektroniikka\Käyttötekniikan automaatio\Palvelut, [Viitattu 2009]. Saatavissa: <http://www.sew-eurodrive.fi/download/pdf/11335416.pdf>. 11335416/EN.

Siemens (2007): Applications&Tools, Technology CPU, "Flying Shears Based on Gearing". [Verkkomanuaali]. Versio 3.2, 06/05/07, Industry Automation and Drive Technologies – Service&Support – Automation Service, A, [Viitattu 2009]. Saatavissa: <http://support.automation.siemens.com/WW/llisapi.dll?func=cslib.csinfo&lang=en&objid=21063352&caller=view>. ID numero: 210633527/EN, kolmiosainen.

Silvola, R. (2006): Reaaliaikaiset teollisuus-Ethernet -ratkaisut automaatiojärjestelmissä. Diplomityö, Tampereen teknillinen yliopisto, automaatiotekniikan osasto. Tampere. 103 s.

Suontausta, I. (2009): Toimitusjohtaja. Iomus Oy. Koikkurintie 4 B 30300 Forssa. Sähköpostihaastattelu, 23.9.2009.

Varis, J. (2009): Professori. Lappeenrannan teknillinen yliopisto. Tuotantotekniikan osasto. PL 20, 53851 Lappeenranta. Sähköpostihaastattelu 27.8.2009.

Varis, J. ja Tanskanen A. (2006): "Terästankojen vannesahaus", Tutkimusraportti 64, Lappeenrannan teknillinen yliopisto, konetekniikan osasto / valmistustekniikan laitos, Lappeenranta, LTY digipaino, ISBN 952-214-212-3.

Voloshkin, J. (2009): Pääsuunnittelija. ABB AS. Valimopolku 4A, 00380 Helsinki. Lentävä leikkaus tapaaminen, 30.10.2009.

Yaskawa (2006): Linear Flying Shear, [Verkkodokumentti, viitattu 2009]. Saatavissa: <http://www.yaskawa.com/site/industries.nsf/applicationDoc/appflyingshear.html>, tuote: MP940/2300, SMC. Dokumentti: AO.MCD.05.110.

Liite A: Kilpailijoilla olevat toiminnallisuudet

Seuraaviin taulukoihin on koottu muiden laitevalmistajien ja sovellustarjoajien toiminnallisuudet lentävä leikkaus -sovelluksessa. Toiminnallisuustarjonta on kartoitettu tutkimalla laitevalmistajien internet-sivustoja ja manuaaleja.

Taulukko A.1. Kilpailijoiden Lenze, Emerson, CT, Siemens ja SEW lentävä leikkaus -sovelluksissa olevat ominaisuudet ja toiminnallisuudet.

	Lenze	Emerson	CT	Siemens	SEW
Leikkaus merkistä	x	x	x	x	x
Leikkaus pituuden mukaan	x	x	x	x	x
Tahdistus masterin nopeuteen	x	x	x	x	x
Ylisynkronointi masterin nopeuteen	x	x	x		
Sysäys	x			x	x
Viivästetty tahdistus	x				
Pituuslaskuri ajometrien summalle	x				
Laskuri leikattujen palojen määrälle, erälaskuri	x	x	x		
Kappalelaskuri(ulkoinen liipaisu)	x				
Skaalauksen yksiköiden		x	x		x
Resoluutio 'leikkauspituudelle', tarkkuus 0,001		x		x	x
Kotiutus	x	x	x	x	
Kotiutuksen poikkeutus					
Laitteiston/Ohjelman rajat	x	x	x	x	
Ohjelman rajakytkinten monitorointi/etähallinta	x		x	x	
Ryömintä	x	x	x	x	x
Virtuaalisen isännän käyttöönotto	x		x		
Absoluuttinen positiointi	x			x	
Leikkausmerkin tarkistus	x				
Rinnakkainen leikkaus		x			x
Profiilin muutokset lennosta		x	x		
Kenttäväylärajoitukset	x	x	x	x	x
Vian käsittely, konfigurointi	x	x	x	x	x
Pyörivä veitsi -sovellus	x				
Nokkapyöräsovellus	x			x	x
Vinoleikkauskulma	x	x	x		x
MMI tuki		x	x		
HMI (Human Machine Interface)	x	x	x	x	x
Löytyneen manuaalin informatiivisuus	***	*	**	***	**

Taulukko A.2. Kilpailijoiden Rockwell, Yaskawa, Baldor, Soloist ja Force Control lentävä leikkaus -sovelluksissa olevat ominaisuudet ja toiminnallisuudet.

Kilpailijalaitteissa olevat lentävä leikkaus -ominaisuudet	Rockwell	Yaskawa	Baldor	Soloist	Force Control
Leikkaus merkistä	x	x			
Leikkaus pituuden mukaan	x	x	x	x	X
Tahdistus masterin nopeuteen	x	x	x	x	x
Ylisynkronointi masterin nopeuteen	x				
Sysäys					
Viivästetty tahdistus					
Pituuslaskuri ajometrien summalle					
Laskuri leikattujen palojen määrälle, erälaskuri					
Kappalelaskuri(ulkoinen liipaisu)					
Skaalauksen yksiköiden				x	
Resoluutio 'leikkauspituudelle', tarkkuus 0,001					
Kotiutus					
Kotiutuksen poikkeutus					
Laitteiston/Ohjelman rajat					
Ohjelman rajakytkinten monitorointi/etähallinta					
Ryömintä					
Virtuaalisen isännän käyttöönotto					
Absoluuttinen positiointi					
Leikkausmerkin tarkistus					
Rinnakkainen leikkaus					
Profiilin muutokset lennosta					
Kenttäväylärajapinnat		x			
Vian käsittely, konfigurointi	x	x			x
Pyörivä veitsi	x	x	x	x	x
Nokkapyöräsovellus					
Vinoleikkauskulma					
MMI tuki				x	
HMI (Human Machine Interface)	x		x	x	
Löytyneen manuaalin informatiivisuus	*	*	*	*	*

Taulukko A.3. Kilpailijoiden Parker, Gefran, Jetter ja Beckhoff lentävä leikkaus -sovelluksissa olevat ominaisuudet ja toiminnallisuudet.

Kilpailijalaitteissa olevat lentävä leikkaus -ominaisuudet	Parker	Gefran	Jetter	Ormec	Beckhoff
Leikkaus merkistä				x	
Leikkaus pituuden mukaan	x	x	x	x	x
Tahdistus masterin nopeuteen	x	x	x	x	x
Ylisynkronointi masterin nopeuteen					
Sysäys		x			
Viivästetty tahdistus					
Pituuslaskuri ajometrien summalle					
Laskuri leikattujen palojen määrälle, erälaskuri					
Kappalelaskuri(ulkoinen liipaisu)					
Skaalauksen yksiköiden		x			
Resoluutio 'leikkauspituudelle', tarkkuus 0,001					
Kotiutus					
Kotiutuksen poikkeutus					
Laitteiston/Ohjelman rajat				x	
Ohjelman rajakytkinten monitorointi/etähallinta					
Ryömintä					
Virtuaalisen isännän käyttöönotto		x			
Absoluuttinen positiointi					
Leikkausmerkin tarkistus					
Rinnakkainen leikkaus					
Profiilin muutokset lennosta		x		x	
Kenttäväylärajapinnat	x	x	x		x
Vian käsittely, konfigurointi				x	
Pyörivä veitsi	x	x		x	
Nokkapyöräsovellus	x				
Vinoleikkauskulma					
MMI tuki				x	
HMI(Human Machine Interface)		x	x	x	
Löytyneen manuaalin informatiivisuus	*	*	*	*	*

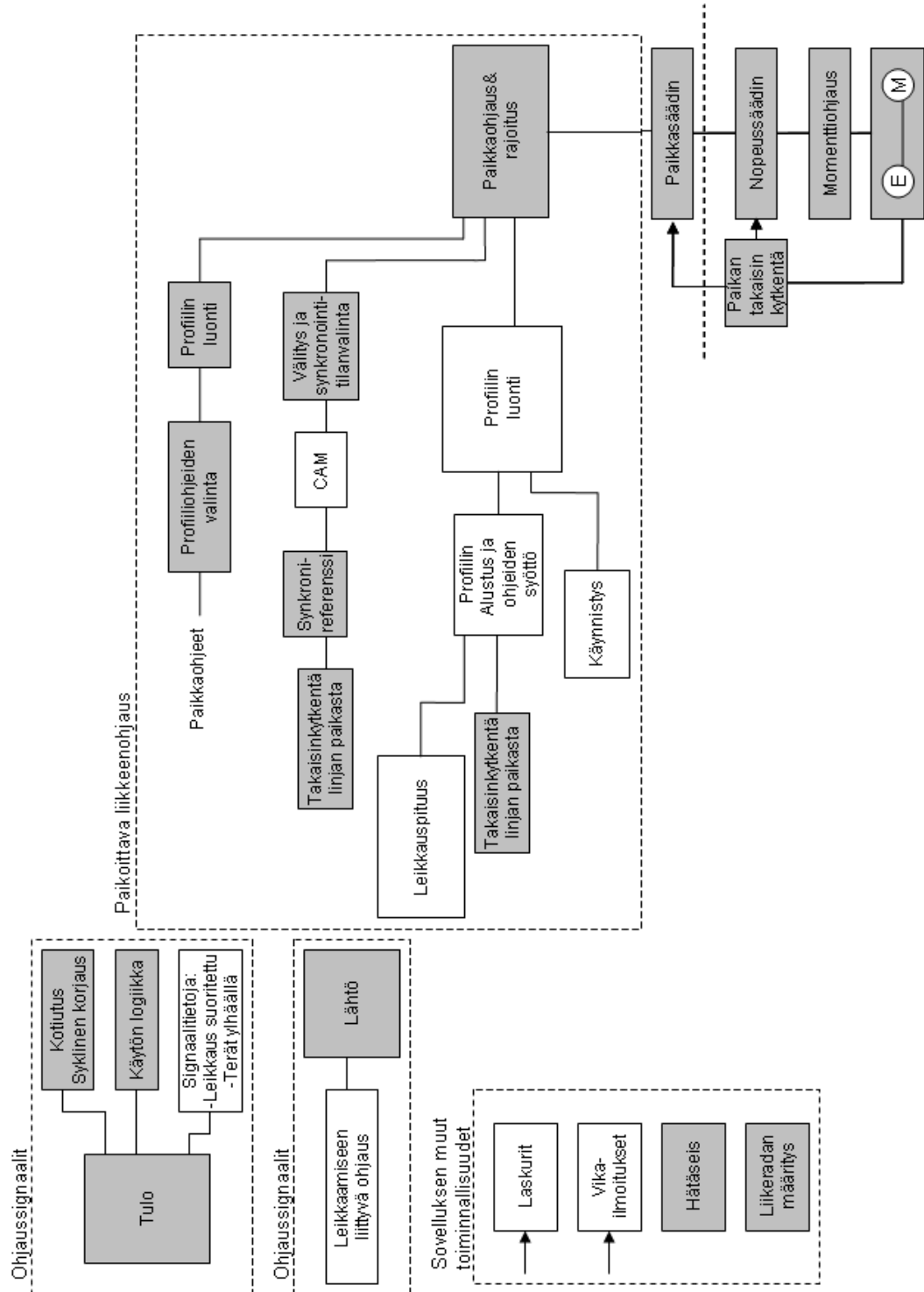
Liite B: Englanninkieliset käännökset toiminnallisuuksille

Taulukko B.1. Englanninkieliset käännökset lentävä leikkaus -sovelluksen ominaisuuksille ja toiminnallisuuksille.

Leikkaus merkistä	Cutting by mark
Leikkaus pituuden mukaan	Cutting by length
Tahdistus masterin nopeuteen	Synchronous synchronization to a master speed
Ylisynkronointi masterin nopeuteen	Oversynchronous synchronization to a master speed
Sysäys	Pulling a gap
Viivästetty tahdistus	Delayed synchronization
Pituuslaskuri ajometrien summalle	Length counter for the total running meters
Laskuri leikattujen palojen määrälle, erälaskuri	Scrap cut counter
Kappalelaskuri(ulkoinen liipaisu)	Scrap counter (externally triggered)
Skaalauksen yksiköt	Scaling units
Resoluutio 'leikkauspituudelle', tarkkuus 0,001	Resolution of the cut-length within 0.001 units
Kotiutus	Homing
Kotiutuksen poikkeutus	Home position offset
Laitteiston/Ohjelman rajat	Hardware/software limits
Ohjelman rajakytkinten monitorointi/etähallinta	Software limit switch monitoring
Ryömintä	Jog functions
Virtuaalisen isännän käyttöönotto	Virtual master enable/disable
Paikoitusvirhe	Following error
Absoluuttinen positiointi	Absolute positioning
Leikkausmerkin tarkistus	Mark checking
Rinnakkainen leikkaus	Parallel cutting
Profiilin muutokset lennosta	Profile changes on fly
Kenttäväylärajapinta	Fieldbus interface
Vian käsittely, konfigurointi	Error handling, configuration
Pyörivä veitsi	Rotary knife
Nokkapyöräsovellus	Cam functionality
Vinoleikkauskulma	Angled carriage applications are handled
MMI tuki	MMI Help
HMI (Human Machine Interface)	HMI (Human Machine Interface)

Liite C: Lentävä leikkaus -säädön toteutusehdotus

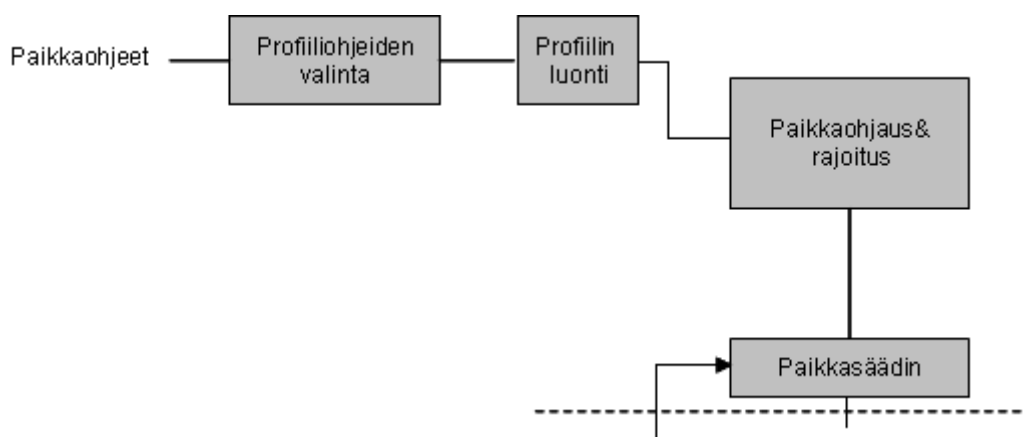
Kuvassa C1 on kuvattu lentävä leikkaus -säätö sekä erikoissäätötilojen kytkeytyminen osaksi nopeus- ja momenttisäätöä sovellusohjelman lohko-ohjelmassa. Mukana on myös muut leikkaussovelluksen toteuttamiseen liittyvät lohkot. Harmaat lohkot kuuluvat ACSM1:n vakiotoiminnallisuuteen. Valkoiset lohkot ovat puolestaan siihen lisättäviä lentävään leikkaukseen liittyviä lohkoja. Seuraavissa kuvissa ja kappaleissa on kuvattu yksityiskohtaisemmin kuinkin osa-alueen toimintaan liittyvät piirteet.



Kuva C1 Arkkitehtuuri peruspaikoituksen, synkronointisäädön ja lentävä leikkaus -säädön toteuttamiseksi sekä näiden kytkeytyminen osaksi muuta liikkeenohjausjärjestelmää.

Paikkasäätö

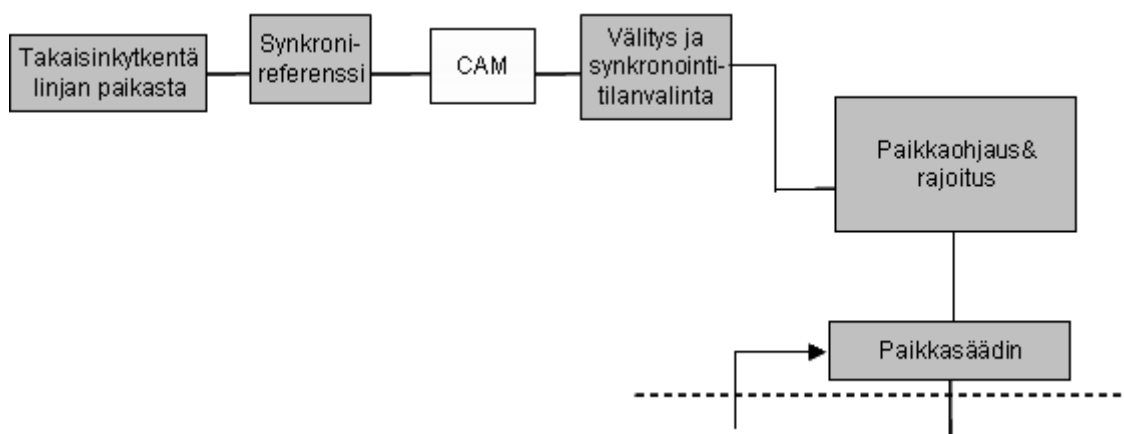
Paikoittavaan liikkeenohjaus -kirjastoon liittyvät erikoissäätötilat jaotellaan kuvassa C1 kolmeen osaan: paikkasäätöön, synkronointisäätöön ja lentävä leikkaus -säätöön. Kuvassa C2 on kuvattu paikoittavan liikkeen ohjaus -kirjaston vakiona sisältämän paikkasäätimen lohkoesitys ja sen kytkeytyminen osaksi perusliikkeenohjausta. Paikkasäädössä kuorma paikoitetaan yksittäisellä akselilla alkuasennosta määriteltyn kohdepaikkaan. Kiinteän ohjelman parametritiedoista annetaan paikkaohjeet kohdepaikan osoittamiseksi.



Kuva C2. Paikoittavan liikkeenohjaus -kirjaston paikoittavan säädön osuus lohkokaaaviona.

Synkronointisäätö

Paikoittava liikkeenohjaus -kirjasto sisältää vakiona myös synkronointisäätötilan, jota käytetään kahden mekaanisen järjestelmän tahdistamiseen. Toiminnaltaan säätö vastaa paikkasäätöä, mutta synkronointisäädössä paikkaohje annetaan liikkuvasta kohteesta, joka lentävän leikkauksen tapauksessa on linjan paikka. Jotta synkronointisäätöä voidaan käyttää lentävä leikkaus -sovellukseen, käyttöön ladattavalla cam-teknologiakirjastolla eli nokkapyörätoiminnallisuudella saadaan epälinearisoitua linjan paikka leikkausprofiilia varten. Cam-lohkon sijainti on esitetty kuvassa C3. Tämä säätötapa ei ole suositeltava tapa sovellusta ajatellen.



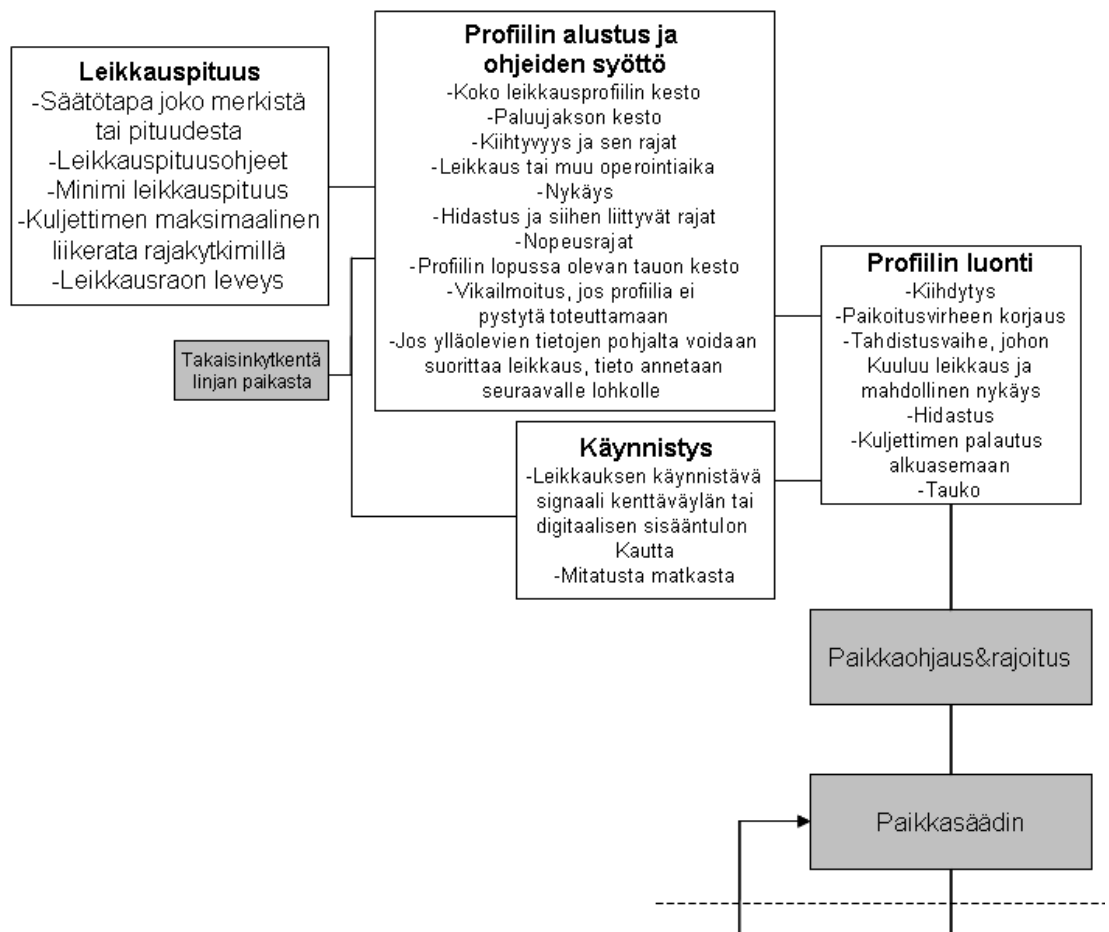
Kuva C3. Synkronointisäätö lohkokaaaviona, ja nokkapyörätoiminnallisuuden (cam) kytkeytyminen osaksi sitä.

Lentävä leikkaus -säätö

Kuvassa C4 on lentävä leikkaus -säädön arkkitehtuuri. Profiilin alustus ja ohjeiden syöttö -lohkoon syötetään kaikki ohjelman tarvitsemat parametrit ja muuttujien raja-arvot. Kaikkia profiilinluonnin tarvitsemia muuttujia ei anneta parametritietoina, sillä tarvittavat loput tiedot lasketaan saatujen parametritietojen pohjalta. Jos profiilin luonti on tietojen pohjalta mahdollista, siirretään tiedot eteenpäin. Muussa tapauksessa annetaan vikailmoitus eikä suoriteta leikkausta.

Lohkoon tulee myös tieto leikkauspituudesta ja takaisinkytkentä linjan paikasta. Leikkauspituustieto voidaan määrittää parametritietona tai kenttäväylän kautta. Siihen lisätään leikkausraon vaikutus, huomioidaan leikkaustapa ja tarkistetaan pituusrajat ennen kuin pituustieto siirretään profiilinluonnin hyväksi. Leikkauspituuden määrittäminen on sijoitettu omaan lohkoonsa, sillä sen määrittämiseen vaaditaan enemmän toimintoja. Takaisinkytkentää linjan paikasta käytetään profiilin luontiin ja kuljettimen paikan oloarvon määrittämiseen.

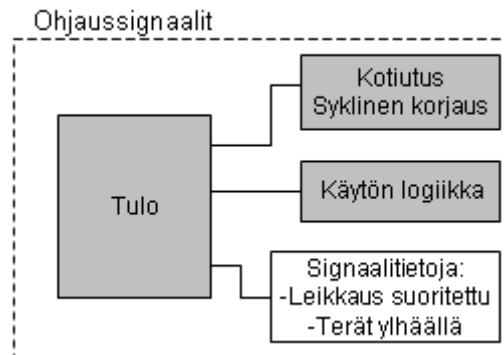
Profiilin luonti -lohkossa leikkausprofiili jaetaan kuuteen perusjaksoon, joita siis ovat kiihdytys, paikoituksen korjaus, leikkaus, jarrutus, palautus ja tauko. Tarvittavien jaksoiden määrä voi kasvaa suuremmaksi leikkauksen erikoistoimintojen myötä. Sellainen voi olla esimerkiksi leikkauksen päätteeksi lisättävä nykäys. Käynnistyslohkosta annetaan käsky aloittaa tai lopettaa leikkaus. Käskyn voi antaa mitatun matkan perusteella, digitaalisen sisääntulon tai kenttäväylän kautta. Lopetuskäsky astuu voimaan, kun käynnissä oleva leikkaus päättyy.



Kuva C4. Lentävä leikkaus -säätö lohkoaviona, lohkoihin liittyvät toiminnallisuudet ja säädön kytkeytyminen osaksi paikoittavaa liikkeenohjausta.

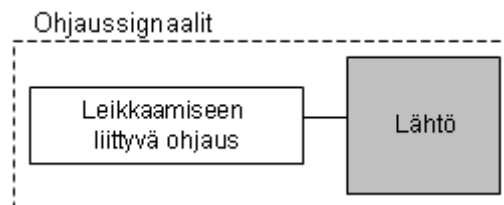
Ohjaussignaalit

Lentävässä leikkauksessa tarvitaan signaalitietoja leikkauksen suorittamisesta sekä leikkauksen käynnistys ja kotiutuskomento vikatilanteiden jälkeen. Kaikki nämä saadaan liikettä ohjaavaan taajuusmuuttajakäyttöön digitaalisena informaationa tai kenttäväylän komentosanoina, kuten kuvassa C5 on havainnollistettu. Kotiutustoiminnot kuuluvat laitteen vakiotoiminnallisuuteen samoin kuin käytön logiikka, joka tarkoittaa käynnistuksen ja pysäytyksen toteutusta. Nämä kuuluvat laitteen vakiotoiminnallisuuteen.



Kuva C5. Sisääntulokytkentöjen kautta saatava ohjausinformaatio.

Leikkausvaiheen aikana tapahtuu leikkaaminen tai jokin muu operointi. Operointikäsky ja muut siihen liittyvät ohjauskomennot on annettava operoinnin suorittavalle järjestelmäkäytön osalle ja niiden siirto tapahtuu ulos käytöstä digitaalisina ohjaussignaaleina tai kenttäväylän komentosanoina. Lohkotasolla ulosmenoa on havainnollistettu kuvassa C6.



Kuva C6. Leikkauskomennot tai muut operointikäskyt lähetään ulosmenevinä ohjaussignaaleina järjestelmäkäytön muille osille.

Muut toiminnallisuudet

Kuvaan C7 on koottu sovelluksessa olevia muita toiminnallisuuksia, joita ovat laskurit, vikailmoitukset, hätäpysäytys ja liikeradan määrittäminen. Näitä ei voida asettaa lohkoesityksessä mihinkään yksittäiseen kohtaan, joten ne on koottu sovelluksen muut toiminnallisuudet osion alle. Laskurille lähetetään laskentasiignaali kenttäväylän tai laskurin vaatimasta toiminnosta riippuen jostakin profiilin luonti -lohkon jaksosta. Laskurilohko laskee saamiaan signaaleja ja pitää kirjaa niistä sekä huolehtii vähennys- ja nollaustoiminnoista.

Vikailmoituslohko voi saada eri paikoista säätöjärjestelmää vikailmoituksia. Jos vikailmoitukset liittyvät lentävään leikkaukseen, ne ohjautuvat tähän lohkoon. Vikailmoituslohkossa viat käsitellään ja kirjataan. Hätäpysäytyskomento pysäyttää taajuusmuuttajan koko liikkeenohjauksen. Tarpeen vaatiessa tähän voidaan vielä liittää joitakin muita toimintoja.

Liikeradan määrittäminen -lohkon toiminnot ovat taajuusmuuttajan kiinteää ohjelmistoa, mutta näiden toimintoja ja informaatiota tarvitaan lentävän leikkauksen toimintoihin, ja sen säätöön. Kotiutuksella kuljetin voidaan siirtää kotiasemaan esimerkiksi vikatilanteen jälkeen ja rajakytkintoimintoja tarvitaan kuljettimen liikeradan määrittämiseksi.



Kuva C7. Sovellukseen liittyviä muita toiminnallisuuksia ja kuvaukset niistä.